University of Warsaw

Faculty of Mathematics, Informatics and Mechanics

Aleksandra Irena Jarmolińska

Student no. 305033

Algorithms and models for protein structure analysis

PhD's dissertation

in COMPUTER SCIENCE

Supervisors:

Dr hab. Joanna Ida Sułkowska, prof. UW

Centre of New Technologies, University of Warsaw

Prof. dr. hab Anna Gambin

Institute of Informatics, University of Warsaw

June 2019

Supervisor's statement

Hereby I confirm that the presented thesis was prepared under my supervision and that it fulfils the requirements for the degree of PhD of Computer Science.

Date Supervisors' signatures

Author's statement

Hereby I declare that the presented thesis was prepared by me and none of its contents was obtained by means that are against the law.

The thesis has never before been a subject of any procedure of obtaining an academic degree.

Moreover, I declare that the present version of the thesis is identical to the attached electronic version.

Date Author's signature

Abstract

Algorithms and models for protein structure analysis

In this work we present several algorithmic approaches designed to help researchers in the study of various orders of protein structure. To facilitate the study of molecular sequence evolution we present an algorithm for multiple alignment of sequence profiles, describe a tool that can be used to study the relationship between residue co-evolution and structure, and a database of structures modeled based co-evolutionary approach. On the structure side, a new algorithm for knot type assignment in biological molecules is introduced, a database of linked protein structures is described, and a method of fixing structure models in a topologically-conscious way is presented. Additionally, folding pathways of several newly discovered knotted proteins are proposed, and the influence of coevolution-based interactions of folding simulations discussed.

Algorytmy i modele do analizy struktur białkowych

Niniejsza rozprawa doktorska omawia szereg metod mających zastosowanie w badaniu białek na wielu płaszczyznach. Pierwszy rozdział wprowadza nowy algorytm pozwalający na określenie typu węzła w biocząsteczkach. Drugi rozdział poświęcony jest ewolucji sekwencji molekularnych. Na początku opisany jest nowy algorytm do multiuliniawiania profili sekwencyjnych oraz jego zastosowanie w badaniu ewolucji białek membranowych zawierających zduplikowane domeny. Następnie przedstawione jest narzędzie pozwalające na badanie związków między koewolucją sekwencji (znalezioną poprzez metodę Direct Coupling Analysis), a strukturą cząsteczki, oraz baza danych struktur wymodelowanych na podstawie koewolucji sekwencji. Wreszcie przedstawione jest zastosowanie oddziaływań wskazanych przez koewolucję w symulacjach zwijania białek. Ostatni rozdział poświęcony jest badaniom nietrywialnych topologicznie struktur białek, poprzez bazę danych struktur zawierających linki oraz metodę naprawy modeli struktur z zachowaniem właściwej topologii. Na koniec przedstawione są propozycje ścieżek zwijania dla nowopoznanych struktur białek z węzłami.

Keywords

Algorithms, Direct Coupling Analysis, Topology, Knots, Evolution, Protein structure

Thesis domain (Socrates-Erasmus subject area codes)

11.3 Informatyka

Subject classification

I.6. Simulation and Modelling
J.3. Life and Medical Sciences

Tytuł pracy w języku polskim

Algorytmy i modele do analizy struktur białkowych

Contents

List of Figures

List of Algorithms

1 Introduction

PROTEINS are traditionally described using three aspects, following the flow of genetic information: sequence, structure, and function. Proteins, also called polypeptides, are biopolymers – molecules, made up of multiple chained amino acids. There are 20 standard proteinogenic amino acids, which differ in physical (size, hydrophobicity, charge) and chemical properties (atomic composition). Each amino acid can be divided into two parts – the side chain (also called residue, and often used interchangeably with "amino acid" when describing a protein), which is distinctive for every type of amino acid, and the main chain, which is identical for all and through which amino acids polymerise. The consecutive bonded main chains of amino acids form the backbone of a protein chain (Fig.1.1).

Sequence of a protein is the order of residues of amino acids that make up its chain, and is also known as protein's primary structure. Based on this sequence, and how different residues interact with each other, the protein folds (either by itself or with help of other proteins) into its final shape[1] – tertiary structure. If more than one protein chain (not necessarily identical) acting in concert is needed to fulfil its designated function they can form a complex, which is then the quaternary structure. Secondary structure, omitted above, are the local structural elements – such as α-helices and β-sheets – which are the base building blocks of more elaborate three-dimensional

[1]Here we disregard any folding errors – misfolds – and disordered proteins.

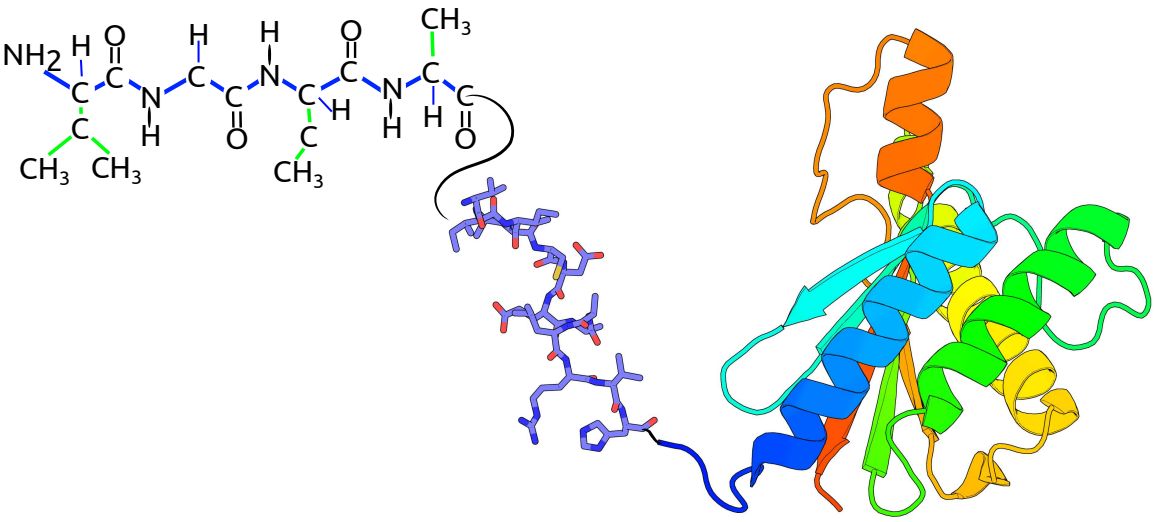

Figure 1.1: Amino acids polymerise into a protein chain. In the molecular formula the backbone atoms are connected by blue lines, side chain atoms by green lines.

configurations, known as folds. As the interactions that guide the proper folding of the protein are the interactions of amino acids that make up its chain, sequence similarity usually indicates a structural similarity (although the reverse is not necessarily true).

Proteins are usually classified based on their primary, ternary, or quaternary structure. The most abundant information available, due to its relative ease and cheapness of obtaining, are the sequences. In general, proteins with similar function, especially if they share a common ancestry, contain one or more similar domains. This allows the use of homologues, that is proteins which descend from the same ancestor protein, as the reference for structure prediction.

This dissertation is focused mainly on proteins, with brief forays into nucleic acids – RNA and chromatin. To study protein structure we apply a broad range of different mathematical tools – from topology and knot theory, through statistics and graph algorithms, up to statistical physics. Figure 1.2 shows a diagram presenting where do they fit in, in the thesis.

1.1. Protein sequence analysis

Protein sequences provide the most information when studied in comparison with others. The order of residues by itself is not yet understood enough to provide much data beyond some basic hydrophobic/hydrophilic differentiation (although this gives some intuition about how structurally buried we can expect a given region to be (Callaway, 1994)). Any further characteristics, such as predicted secondary structure elements, or domain organisation, can only be identified in comparison to already known – extracted from already known proteins structures – statistics (in case of secondary structure prob-

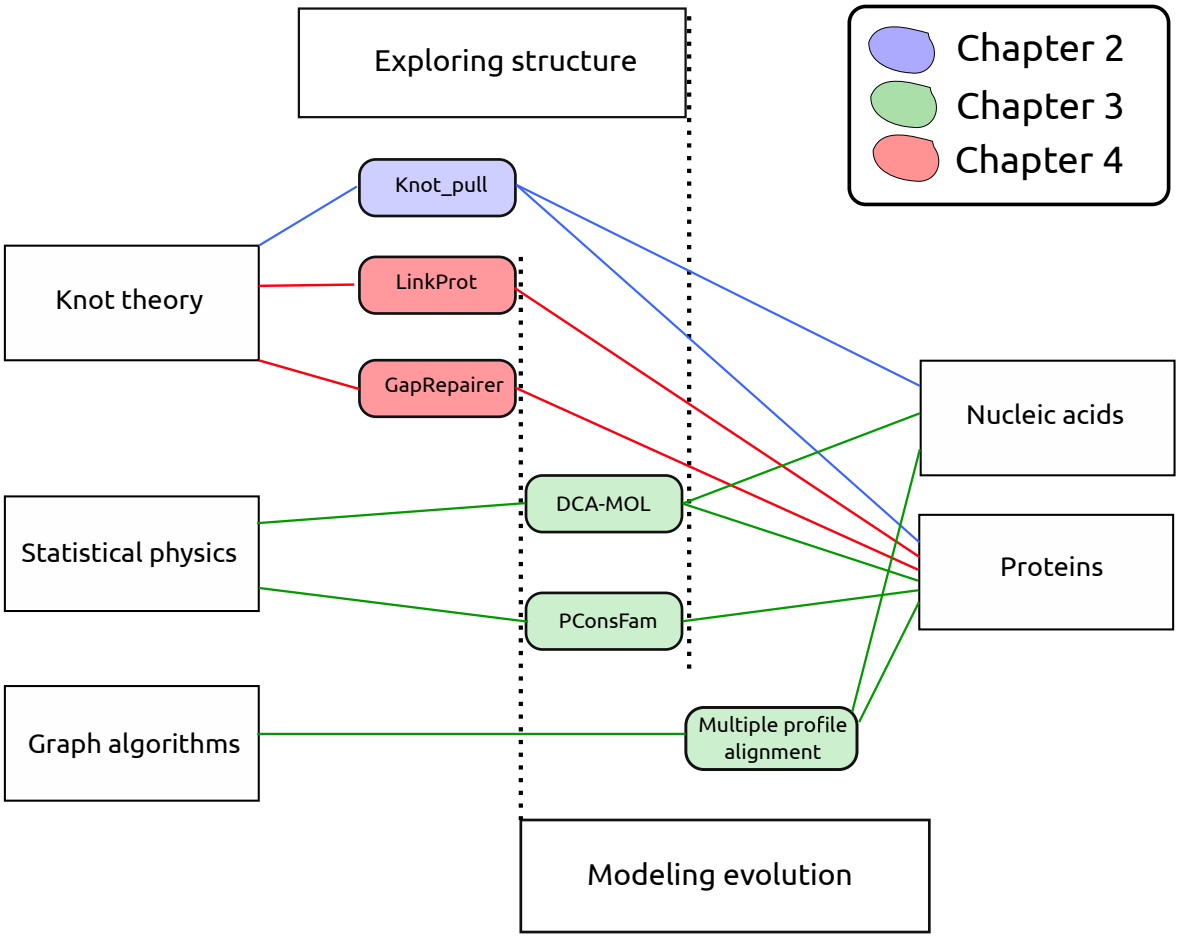

Figure 1.2: Schematic representation of the overlap between various topics included in this thesis.

ability) or patterns (for domains).

1.1.1. Evolution of positions, and of interacting regions

All the evolutionary diversity of phenotypes, considered both on micro- and macroscopic levels, has its roots in just a handful of processes of molecular evolution. Large scale genomic rearrangements – such as gene duplications – are crucial for new proteins to be able to emerge. A duplicated gene has more freedom to mutate, and as long as one of the copies behaves as expected, the other may temporarily (in terms of evolution) lose or change its functions. However, the actual engine of persistent change[2] are the mutations (including insertions and deletions, known together as indels) that affect single nucleotides. When looking at a single nucleotide in a protein-encoding gene, changing the base can start a cascade of changes throughout the central dogma of molecular biology and beyond. If the mutation is not silent – that is the amino acid encoded by the nucleotide triad that changed will be different – the sequence of the

[2]Here we only account for mutations to the genome of the organisms – so those that can be passed on to the offspring.

protein will change. This can result in a direct loss of function, if e.g. the residue was important in ligand binding, or a change to the structure which can then impact the function of the protein or even prohibit correct folding.

Figure 1.3: Exemplary alignment: which the lowest possible edit distance (left), same sequences aligned in a suboptimal way (right).

1.1.2. Sequence alignment

The most straightforward way of comparing two sequences, represented as strings, of same length is to calculate their percentage identity, or some form of edit distance (number of positions which differ). Those metrics can be extended to sequences of different length by finding their 'alignment' which maximizes the identity/minimizes the distance. Alignment here is the association of corresponding positions from each sequence – in practice writing one of the sequences below the others, adding gap symbols signifying indels to one or both, in a way that optimizes the column-wise score (Fig. 1.3). First step in comparing two sequences is an alignment, which finds corresponding residues between them. Alignments are made to satisfy at least one of the following criteria (Claverie and Notredame, 2006):

- evolutionary similarity, where aligned residues hail from the same residue in the ancestral protein;
- structural similarity, where aligned residues are placed in approximately the same position in the protein structure;
- functional similarity, in which aligned residues fulfil the same role in the function of the protein.

For related sequences all three criteria are usually highly convergent, but none can be universally calculated outright from the sequence. Additionally, three sequences with the same number of conserved positions and pairwise identity percentages may still differ in similarity. The assumption that some changes between amino acids are more likely to be successful – or rather mutation between residues with similar properties is less likely to break the protein – lead to the creation of amino acid scoring matrices (such as BLOSUM (Henikoff and Henikoff, 1992) and PAM (Dayhoff *et al.*, 1978)) that indicate how favourable is any given change of residues (with no change being the most favourable, in particular for more peculiar residues). Finding an optimal

alignment is a computationally complex task – for two sequences with lengths M and N it has the time complexity of $\mathcal{O}(N \times M)$ and space complexity $\mathcal{O}(N \times M)$ (can be reduced to $\mathcal{O}(N \cdot \max(1, \frac{M}{\log(N)}))$ (Arlazarov *et al.*, 1970; Masek and Paterson, 1980), and $\mathcal{O}(\min(N, M))$ (Hirschberg, 1975), respectively). Most popular algorithms for sequence alignment are based on dynamic programming, and can be further divided based of their applications (Fig. 1.4):

1. global alignment (e.g. Needleman-Wunsch algorithm (Needleman and Wunsch, 1970)) creates an end-to-end matching between analysed sequences. Can be used to find differences between overall similar sequences. Resulting alignment contains full length of input sequences.

2. local alignment (e.g. Smith-Waterman algorithm (Smith *et al.*, 1981)) finds most similar region(s) of analysed sequences. Typically used to find e.g. shared domains of two proteins.

3. "glocal" (semi-global) alignments (e.g. (Brudno *et al.*, 2003)) attempt to create an alignment where at least one start, and one end of a sequence are present (not necessarily from the same sequence). This is useful when looking for a global alignment of proteins with vastly different lengths.

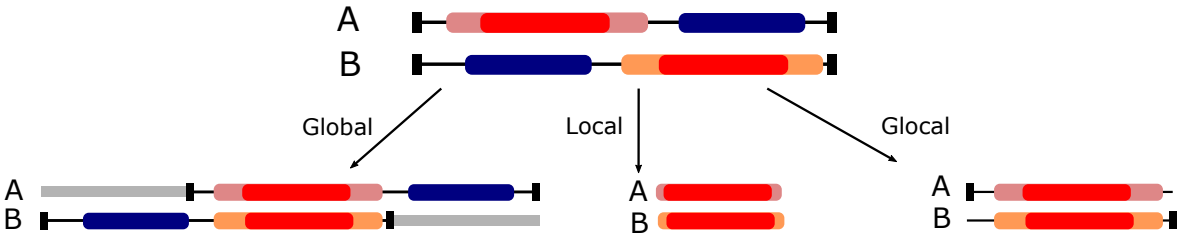

Figure 1.4: Alignment of two protein sequences: (left) global; (middle) local; (right) glocal. Colours indicate regions of similarity.

Other alignment methods include the dot matrix (Gibbs and McIntyre, 1970), which is a binary heat map of identity used to find large scale sequence rearrangements, and k-tuple method which is a sub-optimal solution used for fast database screening. Those approaches are much less versatile than the algorithms mentioned earlier.

It is possible to generalise the pairwise sequence alignment to encompass a larger number of sequences, creating a multiple sequence alignment (MSA). However, only the aforementioned dynamic programming alignment algorithms can be scaled to a larger number of sequences, and those are matrix based (of the size corresponding to one sequence in each dimension). As such calculating them for a real life biological data set is too time-consuming to globally optimise (as it would require creating and finding an optimal path through an N dimensional matrix of size L, where N is the number

of sequences and L is the length of a sequence, assuming it is constant for all). As such, MSAs are commonly calculated using heuristic methods, for example by creating a guide tree based on pairwise distances between sequences, with each node merging the sub-alignments created in its children (Sievers and Higgins, 2014) (see Fig. 1.5). It is worth noting, that usually only global alignment algorithms are used for MSAs, as it possible that some regions of similarity would be discarded from a local alignment before they could be used to match some proteins.

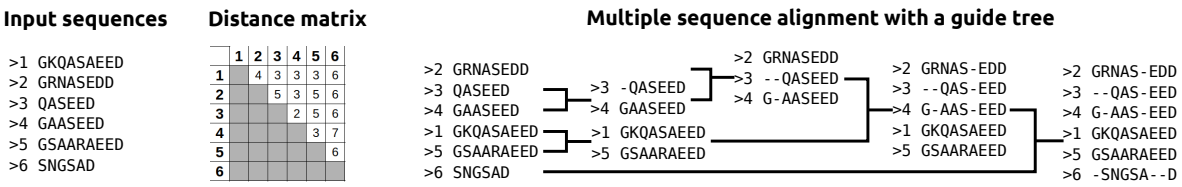

Figure 1.5: Creation of a multiple sequence alignment (MSA) using a guide tree.

Creating an MSA of related, and performing the same function, proteins helps find the residues, or regions, that are conserved through all, or most, of them. In particular, such alignment can then be used to create a sequence profile – a position weight matrix (PWM) resulting from quantitative symbol comparison (Fig. 1.6). Profiles can in turn be used to quantify a similarity of sequence (implicitly – probability of belonging) to a given group, and are in fact the basis of assigning single proteins to protein families (El-Gebali *et al.*, 2018; Finn, 2015). Additionally, profiles (in particular when presented as a sequence logo – Fig. 1.6) provide an easy insight into the entropy of each position, which in turn highlights the best conserved sequence motifs.

Another way of representing sequence profiles is through Hidden Markov Models (HMMs) (Eddy, 1998) – linear state machines, in which there is a *match* node for every column of the alignment (for an ungapped alignment; in alignments with gaps the number of states often can e.g. match the number of columns with less than 50% gaps). Additionally, to handle comparisons with sequences of different length, there should be *delete* states and *insert* states to handle relative indels. Each *match* node should have a set of emission probabilities of generating given symbols, and transition probabilities to its corresponding *insert* node, and the next *match* node and its corresponding *delete* node (as the sequence generation is modeled by a Markov process). HMMs can be trained on a given MSA (e.g. using the Baum-Welch algorithm), and describes the expected pattern behind the known sequences, instead of just the observed counts (as is the case for position weight matrices). As such, HMMs are significantly more sensitive for more remote homology detection (Madera and Gough, 2002).

The most important question to answer which sequence profiles are used is "How likely is a given sequence under the model?", which is used to e.g. to define protein families. Also, there are residues which are more likely to form specific secondary

structure elements and motifs, which makes sequence profiles useful also for some basic structure prediction. Additionally, profiles are much more sensitive than sequences for an alignment, as it is based on column-wise symbol count comparison, and thus doesn't require any specific level of sequence identity to work. However, while there are algorithms for sequence-profile alignments, and pairwise profile-profile alignments, there is currently no method for a multiple profile alignment. In Chapter 3 we introduce a new algorithm that can be used to that end, and describe a phylogenetic analysis which wouldn't be possible without such an algorithm.

MSA

```
>1 GATTAA
>2 CATTAA
>3 GACTAC
>4 CAGTCG
>5 CGATGC
```

PPM

	1	2	3	4	5	6
A	0	4	1	0	3	2
C	3	0	1	0	1	2
T	0	0	2	5	0	0
G	2	1	1	0	1	1

PWM

	1	2	3	4	5	6
A	-∞	1.68	-0.3	-∞	1.26	0.68
C	1.26	-∞	-0.3	-∞	-0.3	0.68
T	-∞	-∞	0.68	2	-∞	-∞
G	0.68	-0.3	-0.3	-∞	-0.3	-0.3

Sequence Logo

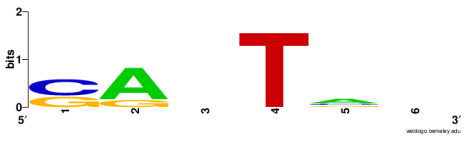

Figure 1.6: From the left: A multiple sequence alignment (MSA) of DNA sequences, corresponding position probability matrix (PWM), position weight matrix (PWM) with no pseudocounts, and sequence logo, which shows bits of information added by each symbol at a given position (Crooks *et al.*, 2004; Schneider and Stephens, 1990).

1.1.3. Sequence-based methods for structure prediction

Protein sequences are much easier and cheaper to discover experimentally, as compared to structures – currently there are more than 150 million sequences deposited in the UniProt (Consortium, 2018) database, three orders of magnitude more than protein structures deposited in the largest structural database – Research Collaboratory for Structural Bioinformatics Protein Data Bank (RCSB PDB)(Berman *et al.*, 2000). However, sequence alone does not provide full overview of the intricacies of a protein. This makes sequence-based structure prediction particularly important. Structure prediction methods can be divided into two main approaches:

- comparative modeling, which uses a known 3D structure as a guide. When available, a structure of a homologous protein is taken as a reference, or, for unrelated proteins, a "threading" of a protein chain in question is attempted in order to recognise a similar fold;
- *de novo* modeling, in which a model is composed either from shown structural fragments known to form from given sequence fragments, a folding simulation is attempted to minimise the energy of a model based on physico-chemical properties of amino acids, or a model fitting some specified contact list is sought (e.g. with contacts found using co-evolutionary methods described below).

Chapter 4 describes GapRepairer – a webserver which uses comparative modeling for a topologically conscious repair of 3D structure coordinates.

1.1.4. Sequence co-evolution

The expanse of the protein sequence space (which for any given length N is 20^N, with 20 being the number of standard proteinogenic amino acids) is far, far greater than actually visited by the evolution regions of functional sequence space (Salisbury, 1969). This highlights how "easy" it is for a protein to mutate and "fall out" from those safe havens. One of the mechanisms that allow proteins sequences to change significantly without losing their function are the compensating or correlated mutations (Taylor and Hatrick, 1994). Change of one residue can severely impact the protein, however if another one mutates to "pick up to slack", either by taking its place in some interaction or e.g. making space to accommodate different side chain (Fig. 1.7), those mutations have a chance to live on. As potential problems associated with sequence mutations are exhibited in structural mishaps, the compensatory mutation should happen close by in terms of spatial proximity. From that we can surmise that if a pair of positions appear to mutate in concert throughout a family of proteins, they are in some sort of contact in the structure.

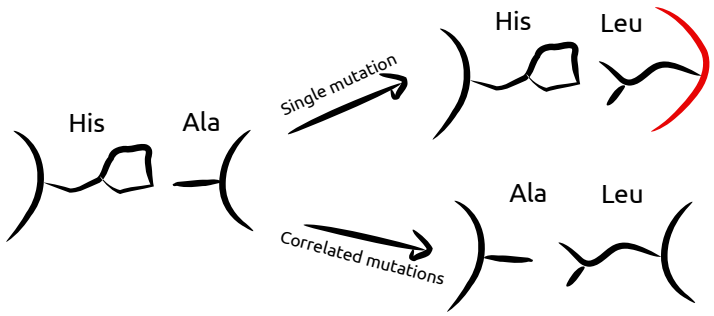

Figure 1.7: An example of a compensating mutation which allows the structure to stay unchanged.

Multiple sequence alignments containing a large set of related sequences can be a source of data for correlation of some positions in the structure, as columns of the alignment should represent evolutionarily linked (in the best case – descendent from the same residue in the ancestral protein) amino acids. Statistical correlation methods (such as Mutual Information (Altschuh *et al.*, 1987; Göbel *et al.*, 1994)) have been used to predict some properties of proteins structures based on MSAs, but more sophisticated approaches have proven to be more discerning of accidental correlations. In particular, a group of methods collected under an umbrella term Direct Coupling Analysis (Morcos *et al.*, 2011; Weigt *et al.*, 2009) have been successfully applied to both protein and RNA structures by disentangling direct correlations from those arising from dependencies on the rest of the sequence.

Chapter 3 goes into more detail on applications of DCA to protein research: Section 3.3.2 presents a tool for easy exploration of DI values in regards to a known structure, Section 3.3.3 describes a database of protein structures modeled based on DCA scores, Section 3.3.4 shows how introduction of DCA-found interactions can facilitate protein folding simulations.

1.2. New order of structure – topology

In 1994 Marc L. Mansfield described a knot-like entanglement he had found in some protein structures, with the "most knotted" that of a human enzyme – carbonic anhydrase (Mansfield, 1994). While he assumed this, very shallow, with one terminus barely passing through a large loop, knotting to be insignificant and a random occurrence caused by thermal fluctuations, he has thus introduced a new descriptor for protein structures. One based on mathematical topology, and as such not fully falling into any of the traditional orders of protein structure – particularly since some of the entanglements found since then are link-like, and so can be defined on a set of protein chains. Next subsections give a brief introduction to mathematical knot theory, and how it was adapted and used for biological molecules. Chapter 2 introduces an improved algorithm for detection and identification of protein entanglements. Chapter 4 describes the creation of some of the online resources which present or use topology of proteins.

1.2.1. Knots in biology

Finding knot-like structures in biological molecules is a non-trivial task – most of the polymers found in biology are open chains, and not closed curves as expected by the mathematical knot definition (descriptions of the strengths and weaknesses of particular knot recognition methods, in particular relating to their application to biomolecules, can be found in Chapter 2). Thus, in general, when a phrase "protein knot" appears, the implicit idea is of a "common sense" knot, as one would find on a string – a structure in which pulling on both ends would not result in a straight line. One notable exception are DNA molecules which can be circular, such as plasmids, and can actually be rather easily knotted and unknotted by topoisomerases (enzymes which sole function is allowing the DNA chain to pass through itself). As such, the circular DNA knots and links have been known and studied for the last four decades (Macgregor and Vlad, 1972; Sumners, 1995).

Much more complex topic are knots on open biological polymers – such as proteins, RNA, and chromatin (open DNA chains). Finding them necessitates a more

lax approach to the mathematical definition, as the curve of the backbones of those molecules must be closed first, to allow the use of tools of knot theory. Typical approach consists of virtually elongating the termini multiple times in random directions, connecting them on a surface of a large sphere around the structure, and calculating the knot type of each resulting closed chain separately. This gives a probability score to each knot type found (an unknot score for Mansfield (Mansfield, 1994)), with the 40% probability cut-off often used to determine the actual presence of a knot (Jamroz *et al.*, 2014).

The most interesting – due to the most diverse nature of chain elements – appear to be proteins. While it can be easily shown, that collapsing a long chain will usually lead to an entangled structure (Levitt, 1976; Némethy and Scheraga, 1977; Skolnick and Kolinski, 1991; Chan and Dill, 1993) and this will in fact represent the most advantageous packing of the polymer, protein folding have been long thought to elude this tendency (Bryant *et al.*, 1974). While folding is generally thought to be guided by hydrophobic collapse, various interactions (both attractive and repulsive) between amino acids that make up the protein chain complicate this process – in particular, a lack of reptative motions of the chain is expected (as the protein chain is not a smooth one). Because of this, the knotting should happen close to its native position in the structure – giving the protein entanglements another possible classifier: their depth. This depth is defined as the minimum number of residues have to be removed from either end of the chain to untangle it. An often complimentary quantity is "tightness" of a knot, which describes how long is the knot core – minimal knotted fragment of a structure.

Due to the fact that the entanglement cannot slide along the protein chain, the rate limiting step in formation of a knot-like structure is assumed to be the piercing of the loop. Consequently, all the protein entanglements found to date correspond to knots which can be made by just a single passing of the chain through a (potentially multiple times) twisted loop (Sułkowska *et al.*, 2012b; Taylor, 2007). For example – while a 5_2 knot-like structure can be found in a family of deubiquitinases, there are no known structures similar to the 5_1 knot (see Fig. 2.2).

In the years following Mansfield's discovery of a shallowly knotted anhydrase, the number of knotted protein structures increased significantly, starting with the discovery of first deeply knotted protein by Taylor in 2000 (Taylor, 2000). To date, entanglements have been found in approximately 1% of known protein structures, with the simplest non-trivial knot (3_1) being the most common (Jarmolinska *et al.*, 2019a). The largest family of knotted proteins are the SPOUT methyltransferases (Tkaczuk *et al.*, 2007; Lim *et al.*, 2003). It is important to note, that entanglements appear to run in the

family (Sułkowska *et al.*, 2012b) – the only known example of a protein family with members differing in topology are the ATCase/OTCases, which groups two related enzymes: aspartate (knotted) and ornithine (trivial) carbamoylotransferases. Entanglement often coincides with the active site (Tkaczuk *et al.*, 2007), but its actual function remains unknown – if there even is a one common denominator of protein knotting. Entanglement has also been linked to increased stability of a protein – both in terms of environmental influences (such as high temperatures) and intracellular degradation machinery (Mallam *et al.*, 2010), as well as mechanical resistance (Sułkowska *et al.*, 2008; Sriramoju *et al.*, 2018). Some studies (Mallam *et al.*, 2008) have shown that entangling the chain is a rate limiting step in folding (including, due to high potential of errors) – and thus can be involved in metabolic regulation, with knotted proteins, slower to fold and slower to degrade, having a lower turnover. The most complex protein knot found to date is the Stevedore (6_1) knot in a bacterial haloacid dehalogenase (Bölinger *et al.*, 2010).

Knotted proteins can be found in organisms from all branches of the Tree of Life, and in molecules with vastly different functions – from enzymes (including mitochondrial ones), through membrane proteins, up to ribosomal subunits (Jarmolinska *et al.*, 2019a). Knots themselves vary in tightness – from cores of less than 50, up to almost 500 residues – and depth – from shallow knots, with tails made up of a few residues, to a bacterial protein which has a whole domain on each of the knot's tails.

Other types of entanglements which can be found on the backbone trace of a protein (Fig. 1.8) include "slipknots" (where the twisted loop was pierced through another loop – thus only a subchain of the protein is knotted) and links (which are again an open chain variation on the mathematical topology). Both have been found and classified by methods similar to those employed in search of protein knots, either by applying them to fragments of the chain, or multiple interacting chains (Jamroz *et al.*, 2014; Dabrowski-Tumanski *et al.*, 2016a).

Chapter 4 explores in more detail the diversity of entangled proteins – Section 4.1.1 presents a database of protein structures containing links, Section 4.3 describes possible folding pathways for some of the newly discovered knotted protein structures.

In case of RNA molecules, screening of the RCSB Protein Data Bank has found only a handful of entangled structures (Micheletti *et al.*, 2015), however none have held up to closer scrutiny – they appeared to be structure determination errors due to low resolution, as all their homologues were found to be unknotted.

One of the challenges not yet overcome by structural biology is the structure of chromatin (Mirny, 2011). Compared with proteins, packed chromosomes are about two orders of magnitude larger, comparatively much more densely packed, and on the

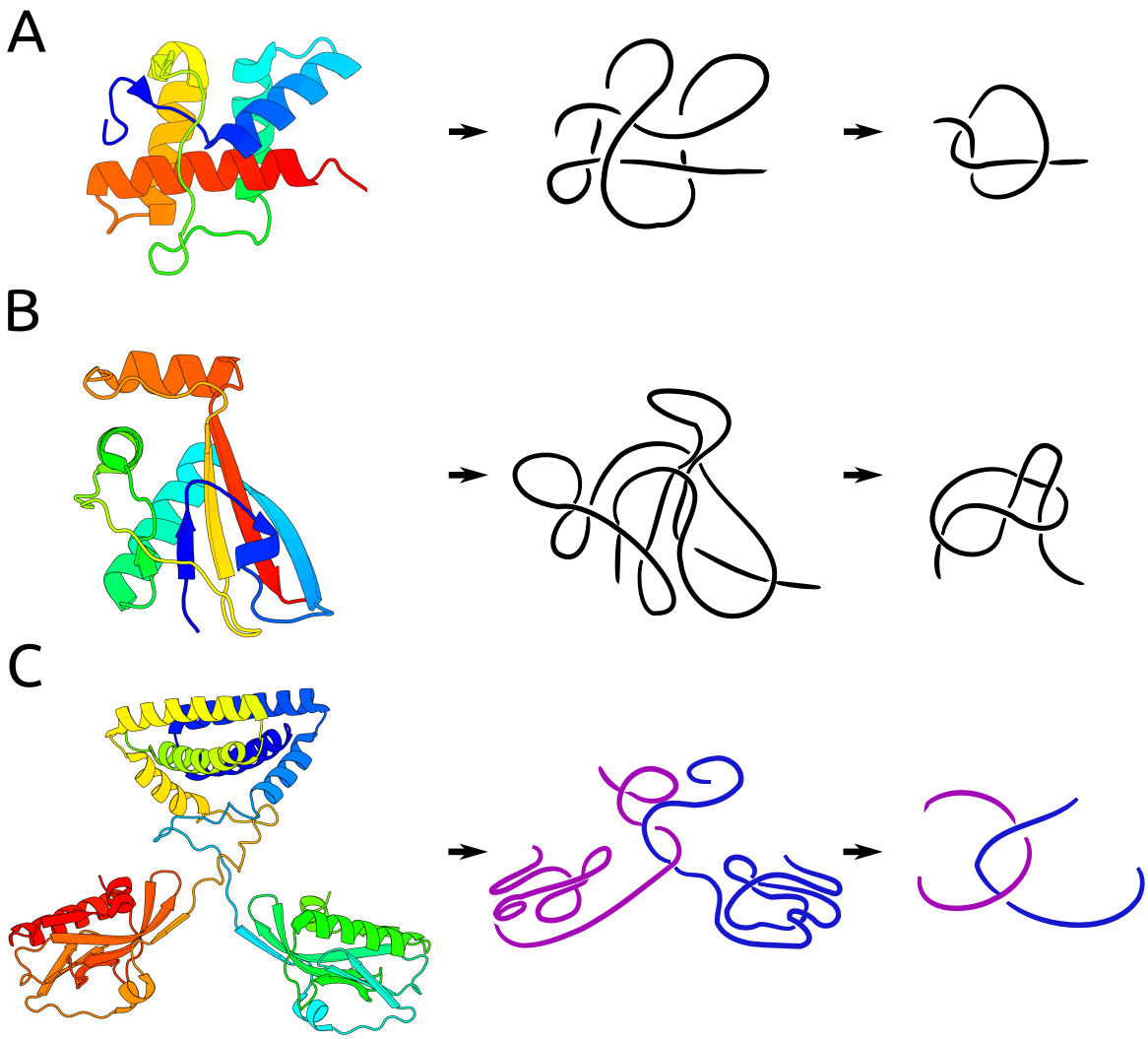

Figure 1.8: Structure, topology sketch and a simplified topology skech of: (A) a knotted protein (PDB Id 4rlv chain A); (B) a slipknotted protein (PDB Id 2j85 chain A); (C) a linked protein (PDB Id 5nt2 chains D and E). Structure visualisations made using PyMOL.

whole don't have a stable conformation. All of the above make traditional methods of structural biology unsuitable for chromatin. Currently, the most successful studies involve 3D models based on contact matrices obtained from Hi-C experiments (Stevens *et al.*, 2017). Studying the topology of the chain can be an important resource in verification of the structures (as excessively entangled chromosomes would prove difficult for the cell to separate during division), or could provide unexpected insights into the packing of biomolecules (as simple polymers do tend to entangle when collapsed). Recently proposed models of mouse chromosomes (Stevens *et al.*, 2017) have been found (Sulkowska *et al.*, 2018) to contain composite knots, which would be a first in biomolecular topology.

Knot_pull algorithm, presented in Chapter 2, provides the first biomolecular tool for deterministic detection of such structures.

1.3. Our contributions

In this thesis we present our contributions to the field of protein science. For the research into the molecular evolution of proteins sequences, we introduce the first multiple profile alignment algorithm, and use it in the first study into the evolution of slipknotted topology in membrane proteins. We give describe several approaches to studying the relationship between structure and sequence co-evolution. For the study of protein topology we introduce new algorithms for structure simplification and knot type assignment. Finally, we present a database of proteins with linked chains, a server for topologically-conscious repair of structure models, and propose folding pathways for several newly discovered knotted proteins.

Publications included in Chapter 2

Jarmolinska, A. I., Gambin, A., Sulkowska, J. I. (2019). Knot_pull - python package for biopolymer smoothing and knot detection. Bioinformatics *(under review)*

Publications included in Chapter 3

Jarmolinska, A. I., Zhou, Q., Sulkowska, J. I. and Morcos, F. (2019b). Dcamol: A pymol plugin to analyze direct evolutionary couplings. *Journal of Chemical Information and Modeling*, **59** (2), 625-629.

Lamb, J.*, **Jarmolinska, A. I.***, Michel, M.*, Menéndez-Hurtado, D., Sulkowska, J. I. and Elofsson, A. (2019). Pconsfam: An interactive database of structure predictions of pfam families. *Journal of Molecular Biology*, **431** (13), 2442-2448.

Dabrowski-Tumanski, P., **Jarmolinska, A.I.** and Sulkowska, J. I. (2015). Prediction of the optimal set of contacts to fold the smallest knotted protein. *Journal of Physics: Condensed Matter*, **27** (35), 354109.

Publications included in Chapter 4

Jarmolinska, A. I., Kadlof, M., Dabrowski-Tumanski, P. and Sulkowska,J. I. (2018). GapRepairer: a server to model a structural gap and validate it using topological analysis. *Bioinformatics*, **34** (19), 3300-3307.

Jarmolinska, A. I., Perlinska, A. P., Runkel, R., Trefz, B., Ginn, H. M., Virnau, P.and Sulkowska, J. I.(2019). Proteins' knotty problems. *Journal of Molecular Biology*, **431** (2), 244-257.

Dabrowski-Tumanski, P.*, **Jarmolinska, A. I.***, Niemyska, W.*, Rawdon, E. J., Millett, K. C. and Sulkowska, J. I. (2016). Linkprot: A database collecting information about biological links. *Nucleic Acids Research*, **45** (D1), D243–D249.

Other Publications

Sulkowska, J. I., Niewieczerzal, S., **Jarmolinska, A. I.**, Siebert, J. T., Virnau, P. and Niemyska, W. (2018). Knotgenome: a server to analyze entanglements of chromosomes. *Nucleic Acids Research*, **46** (W1), W17-W24.

Acknowledgements

I would like to thank all the people that contributed to the creation of this thesis.

I would also like to thank the National Science Centre for their financial support and entrusting me with grant number 2018/29/N/NZ2/02897. The papers described in this work have also been financed by the National Science Centre grant 2012/07/E/NZ1/01900 SONATA BIS to JIS, and Polish Ministry of Science and Higher Education grant 0003/ID3/2016/64 Ideas Plus to JIS.
Also, I'd like to thank Marta Wiśniewska for keeping me from getting a big head by always reminding me of all my board game defeats. And Mariusz Lachowicz for being the good guy who doesn't do that.
Finally, I would like to thank Tali, Makepeace and Higgs for mizi.

Agacie, bez której nie powstałaby nawet molekuła tej pracy.

Głupcy wiążą węzły. Mędrcy je rozwiązują.

Mądrość ze zrywanego kalendarza dla gospodyń domowych

We may be unable to define a knot in an open path, but we know one when we see it.

Marc L. Mansfield, paraphrasing Justice Potter Steward

2

Detection of knot-like folds in biological molecules

By mathematical definition, knots are closed curves in a 3 dimensional space. Because of that, most biological molecules[1] can never be described as mathematically knotted – as the examples of closed backbones of biomolecular chains are few and far between (the only naturally occurring example of cyclic peptides are cyclosporins). Even disregarding the openness of the chain, even a very entangled curve is always isomorphic to a straight, unknotted line. However, this approach disregards the complexity of interactions that must be involved in knotting e.g. a protein.

To this end we can consider "common-sense" knots – chains where one of the ends pierces through a twisted loop. These, while possibly less interesting from the mathematical point of view, have nonetheless real consequences for the molecule in question. In Chapter 4 we describe different web servers and databases of knotted proteins and chromatin models, all of which follow the methodology introduced in first papers on knotted proteins (Mansfield, 1994; Taylor, 2000). This chapter is dedicated to a new algorithm we designed for recognising the topology of open polymers. Implementation is available on GitHub (`http://github.com/dzarmola/knot_pull`).

Chapter is divided into following sections:

[1] with the notable exception of circular DNA, which is known to be *mathematically* knotted

1. Knots in knot theory, a short introduction into the mathematical study of knots;

2. Knot type assignment, which details the application of Dowker-Thisthelthwaite code to knot recognition, and an algorithm for its reduction;

3. Smoothing the chain, which describes an algorithm for topologically-conscious reduction of 3D coordinates of a polymer, as well as its application for link detection;

4. Validation of results, which contains the comparison of the topology assessment using knot_pull with available resources on biomolecular entanglements.

Knot_pull is a new tool for analysing the topology in open chains – such as proteins, RNA and DNA (chromatin). It was designed to bypass some of the particularities of currently used methods.

Presently, most software (Tubiana *et al.*, 2018; Lua, 2012; Jamroz *et al.*, 2014) uses the same basic approach:

1. smoothing (simplification) of the chain, to get a curve with the same overall topology, but less crossings in a plane projection;

2. closing the curve on the surface of a large (implicitly – infinite) sphere surrounding the structure – this step can be error prone, as the closures could add additional entanglement to the chain. Thus such closure is usually repeated multiple times, to lessen the impact of occasionally erroneous path.

3. Closed chain is projected on a plane, and a knot invariant (e.g. Alexander polynomial (Alexander, 1928), or HOMFLY-PT polynomial (Section 2.0.1, Equation 2.1)) is calculated. For multiple closures this gives a probability that the structure contains a given type of a knot.

2.0.1. Knot theory

Mathematically, knots are embeddings of a closed curve (simply put, a circle) in three dimensional Euclidean space. Knots can be classified according to their complexity, expressed as the number of crossings (or double points) in their projection on a surface (a diagram).

Definition 1 (Link diagram)**.** *An orthogonal projection of a knot or link into a plane that contains a finite number of multiple points (double points with transverse crossing), gives a link diagram $\mathcal{D}$ – an undirected labeled planar graph, which satisfies following conditions:*

1. *loops are connected components with no vertices (thus disconnected from the rest of the graph);*

2. *both ends of a non-loop edge lead to one vertex each (possibly the same one), and both are labeled as undercrossing or overcrossing in corresponding vertex;*

3. *there are two overcrossing, and two undercrossing, incident edges leading to each vertex, and their cyclic ordering alternates under- and overcrossings.*

A knot diagram is a link diagram with only one connected component. Vertices of a link diagram are called crossings.

To simplify matters all knots which are isomorphic, that is can be twisted into one another without the "strings" that make them up passing through themselves, are classified according to the minimal number of crossings they can be morphed to contain (Fig. 2.1). The simplest knot, also called an unknot, is a circle which does not contain any crossings. Next in order of complexity is a trefoil (a 3_1 knot in the most common, Alexander-Briggs (Alexander, 1928), notation) with three crossings. For 5 and more crossings there exist multiple non-isomorphic knots with this number of crossings, and they are differentiated through a lower index.

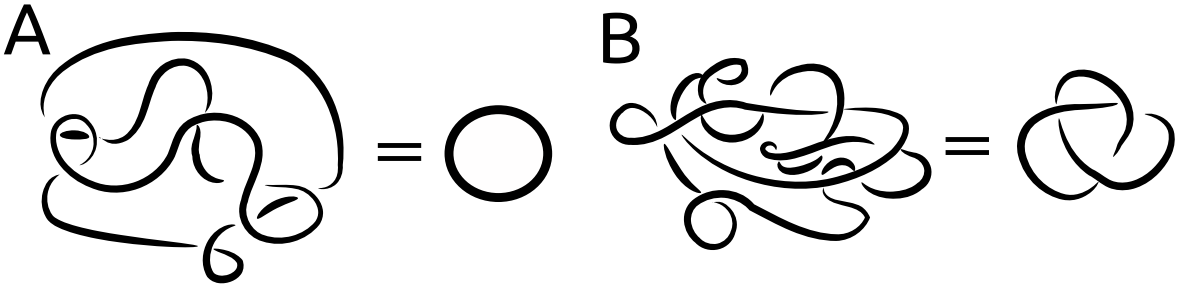

Figure 2.1: A. Exemplary knot diagram isomorphic to an unknot. B. Exemplary knot diagram isomorphic to a trefoil knot.

A collection of non-intersecting, but potentially linked together chains is called a link. Two (or more) oriented knots can be summed by connecting their planar projections by straight bars in such a way that their orientation is preserved. This connected sum (Rolfsen, 1976), for non-trivial knots, is called a composite knot, and allows the definition of *prime knots* – knots that cannot be decomposed into non-trivial simpler knots. As an example – there are three possible non-equivalent prime knots with six crossings, and two non-equivalent composite knots with the same number of crossings (a sum of two trefoil knots). Knot tabulations, which are used to classify and relate different knot notations, usually include only prime knots.

First knot tabulation has been attempted by Peter Tait in 19^{th} century and contained mostly alternating knots – knots in which diagrams consecutive crossings along the chain alternate in orientation (going over/under).

An interesting, in terms of the process of entangling, group of knots are twist knots. They are obtained by joining the ends of a twisted loop (with one of the ends passing through the surface of the loop) – i.e. creation of a twisted knot on a string needs only one "threading action" performed by one of the ends. A half-twisted loop results in a 3_1 knot, then adding consecutive half-twists yields a 4_1, 5_2, 6_1 etc (Fig. 2.2).

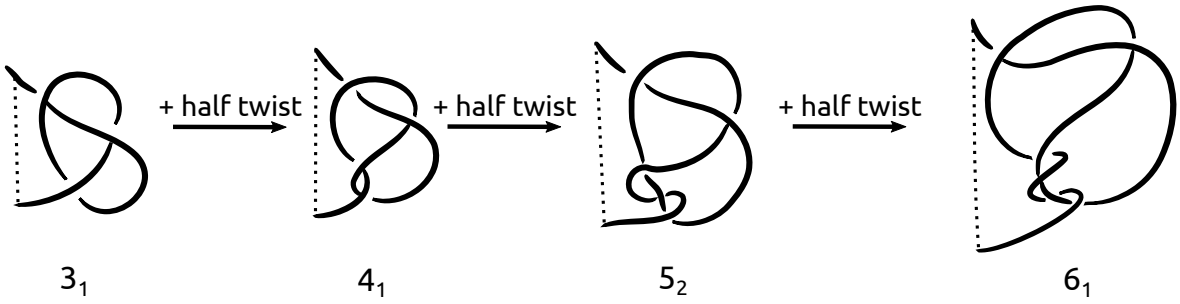

Figure 2.2: Twist knots are knots that can be made with just one threading through a twisted loop.

Additional characteristic that can be used to differentiate knots for which a direction of the curve has been specified is chirality. It is usually symbolised by a $+$ or $-$ sign in the notation, and separates mirror images of a knot. However, some of the knots are achiral (e.g. 4_1 knot).

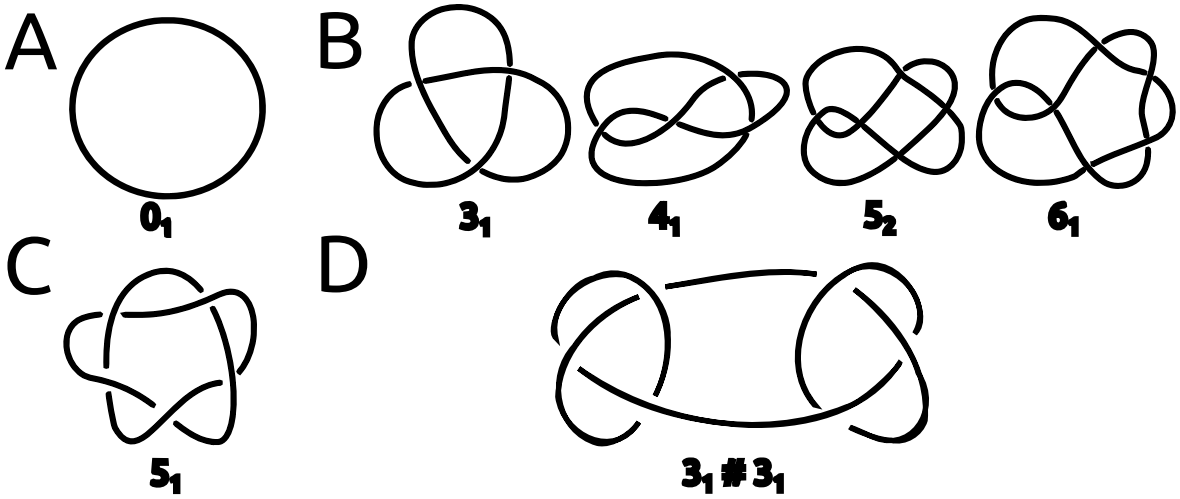

Figure 2.3: Examples of mathematical knots and their Alexander-Briggs notation. (A) The trivial knot – an unknot. (B) Knots which analogues have been found in proteins (with 3, 4, 5 and 6 crossings). (C) A knot with 5 crossings for which no protein analogues have been found. (D) A composite knot – made up of two prime (in this case both with 3 crossings) knots.

Usually knots are recognised through knot invariants. Simply put, a knot invariant is any quantity which can be defined for any knot, and is the same for equivalent (i.e. isomorphic) knots. However, it is important to note that the reverse does not hold – that is dissimilar knots can have the same invariant (e.g. in case of Alexander polynomial a prime, 8-crossing knot as the same invariant as a composite 6-crossing knot). Most commonly used knot invariants are knot polynomials (Alexander, 1928),

which are computed on a given knot diagram (although the choice of a diagram is irrelevant, as per the definition of the invariant). Examples include Alexander polynomial (Alexander, 1928), Jones polynomial (Jones, 1985) and HOMFLY-PT polynomial (Freyd *et al.*, 1990)(the last one used also for links). Polynomial coefficients are calculated through manipulation of crossing directions (e.g. using skein relations), and, as invariants, encode some properties of a knot.

The HOMFLY-PT polynomial is a generalisation of the Alexander and Jones polynomials, and can be transformed into both given appropriate substitutions. The polynomial is defined using skein relations (Fig. 2.4), which give a linear relation between knot polynomial values for links which differ by one crossing (skein relations are sufficient to calculate Alexander and Jones polynomials just by recursion).

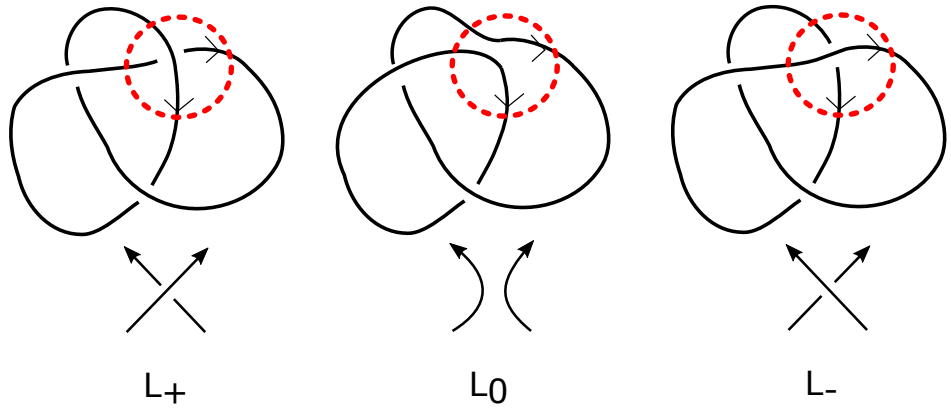

Figure 2.4: Skein relationship can be defined for three link diagrams differing in one crossing. Each diagram should have the crossing in question in different configuration, possible crossing diagrams are shown.

With link diagrams L_-, L_+, L_0 as pictured in Fig. 2.4, the HOMFLY-PT polynomial can be defined as follows:

$$P_U(l,m) = 1 \tag{2.1}$$
$$lP_{L_+}(l,m) + l^{-1}P_{L_-}(l,m) + mP_{L_0}(l,m) = 0,$$

where U is an unknot, and l and m are polynomial coefficients used to distinguish the invariants of different knots. Some main properties of this polynomial include:

- HOMFLY-PT polynomial of a composite knot is the product of polynomials of its components;
- HOMFLY-PT polynomial can be used to distinguish two knots of different chirality, as $P_K(l,m) = P_{Mirror\ image(K)}(l^{-1},m)$

An example of a knot notation which, when properly simplified, gives the actual number of crossings is the Dowker (also called Dowker-Thistlethwaite – DT) code (an improvement on the Tait code) (Dowker and Thistlethwaite, 1983). To obtain this

notation for a surface projection of a knot, we start at some arbitrary point of the curve and while moving along the string we number each crossing that we pass. For a proper knot when we get back to the starting point each crossing should be labeled by two values – one odd and one even (Fig. 2.5). To also take into account chirality of the structure, the crossings can be additionally marked to indicate whether we pass under or over the second line when crossing (which is indicated by a minus sign next to the even value if we add it when going over). The notation can then be shortened, by sorting the label pairs according to the odd value – final order (and positive/negative sign) of the even numbers indicate the knot type. However, it should be noted that this is not an invariant – as one knot diagram can have multiple different notations.

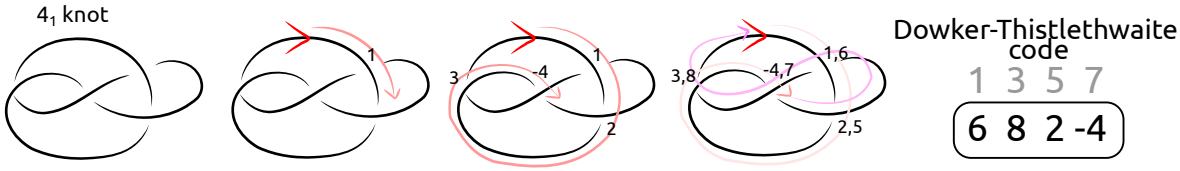

Figure 2.5: Dowker-Thistlethwaite code annotation of a 4_1 knot diagram.

2.1. Knot type assignment

The most problematic step is the selection of a closing path, as due to the diverse packing of molecular structures selection of universally minimally interfering closure is non-trivial (Tubiana *et al.*, 2018). However, polynomial based knot detection makes this step mandatory, as the polynomials are only properly defined for closed curves. Knot_pull bypasses this hurdle, by using the Dowker-Thistlethwaithe (DT) code for topology assignment.

By definition, DT code is assigned based on the plane projection of a 3D structure, as follows: starting from an arbitrary point on the curve, follow the chain (in, again, an arbitrary direction) until you get back to the starting point. Along the way, number all crossings as you pass them, noting whether you were going over or under the crossing. When this is finished, all crossings should be annotated with two numbers (corresponding to two chain segments making up each crossings), one of them marked as going on top of the other. Additionally, it was shown in (Dowker and Thistlethwaite, 1983) that in a properly created code one of the numbers in each crossing will be odd, and one even. This notation is not a widely used one in knot theory, as it cannot be used as a knot invariant. Due to the flexibility of the starting point and movement direction selection, same knot can have different codes. This however can be seen as a source of additional information when studying open polymers, as the starting point (the N terminus) and direction are **encoded** in the structure of a protein. This in turn

makes it easier to differentiate structures with different realisations of a given knot type (see Fig. 2.6). Additional advantage of using the Dowker-Thistlethwaite notation is that the code for a closed chain is the same as the code for an open chain with the break in the corresponding starting point. As a result, the code assigned to the open chain can be said to represent a closed chain, with the implicit closing path that does not contribute any crossings. If no such path is possible, the resulting code will be incorrect (that is there will be crossings with two numbers of the same parity) – which indicates that a different plane projection should be used.

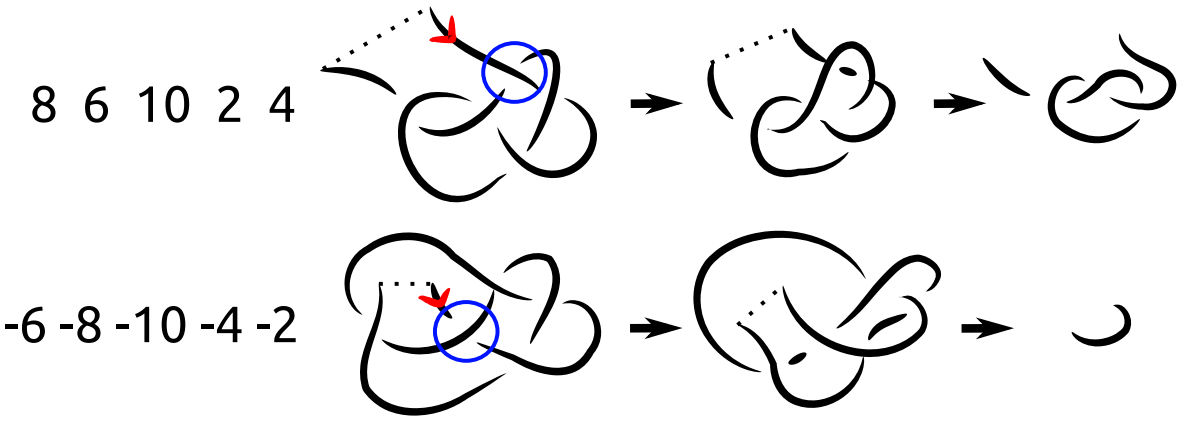

Figure 2.6: Different realisations of an open chain 5 crossings knot (analogous to the 5_2 knot) with their Dowker-Thistlethwaite code. Removing (inverting) the crossing indicated by a blue circle leads to: (left) a 3 crossings knot; (right) an unknot.

On the contrary to knot polynomials, DT code describes a particular projection – even with the same starting point and direction, adding a twist to the chain will result in a change in notation. Since this in principle results in an infinite number of descriptions for a given knot type, knot_pull calculates the shortest DT code. This is done by through a series of code reductions based on the Reidemeister moves of the chain (which have been shown to be sufficient to relate knot diagrams with the same knot type (Reidemeister, 1927)).

First, let us define the DT code as a list of N crossings $DT = \{C_1, C_2, C_3, ..., C_N\}$, each crossing C_i comprised of two values $C_i.a, C_i.b$. Then the potential notation simplifying modifications can be found as follows:

1. Reidemeister move type I – untwisting:

   - a simple loop (a crossing annotated with two consecutive values $|C_i.a - C_i.b| = 1$) can be removed with no influence on the rest of the structure (Figure 2.7 A)
   - when another segment passes through a simple loop ($|C_i.a - C_j.b| = 2; C_i.a < \{C_j.a, C_k.a\} < C_i.b$ and the chain goes sequentially through C_i and C_j on

the same side (over/under), different than through C_k) the twisted crossing can be removed but the order of "internal" crossings along the strike-through segment must be reversed (Figure 2.7 B);

2. Reidemeister move type II – moving a loop from over/under another:

   - if two crossings are consecutive twice ($|C_i.a - C_j.a| = 1 \wedge |C_i.a - C_j.b| = 1$, and consecutive values are both/neither marked as going on top) they can be removed (Figure 2.7 C)
   - if the chain has the same overhandedness in two consecutive (along one value) crossings, they can possibly be replaced by one new crossing – this is verified by trying to find a second value (first being one of those consecutive) that would give a valid DT code – if no such value can be found, no simplification is made (Figure 2.7 D);

3. Reidemeister move type III – moving a segment completely over/under another crossing: this is the only move implemented which does not explicitly simplify the code, however it is attempted when no further simplification can be made (on condition that it will give a code not yet encountered).

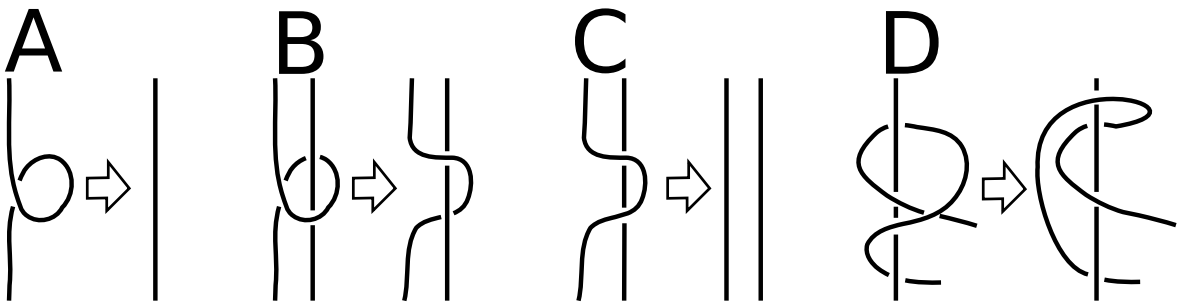

Figure 2.7: Visualisation of code simplifying moves implemented. (A,B): based on Reidemeister move Type I. (C,D): based on Reidemeister move Type II.

Lemma 1. *Let N be the initial number of crossings. If the third Reidemeister move is not attempted, then the time complexity of the Dowker-Thistlethwaite code reduction scheme introduced here is $\mathcal{O}(N^2)$.*

Proof. In each pass of the reducing loop, we iterate over each crossing to check if it can be somehow reduced, which gives the outer N term. This loop will be run up to N times (inner N term), with at least one crossing removed in each pass, until no further simplification can be made. ■

The third Reidemeister move is used if after simplifications any crossings remain, then all the segments which can be safely (without introducing additional crossings)

passed over/under a crossing are moved one by one until the chain can be simplified further, or all configurations with different DT code have been explored (combinatorically at most $N!$, however this number would be never explored, as most are quickly reducible).

Problem 1 (Unknotting problem)**.** *Given a link diagram $\mathcal{D}$, is $\mathcal{D}$ a knot diagram representing a trivial knot?*

Remark. *Determining if the knot diagram represents an unknot has been shown to be in co-NP class (Lackenby, 2016) and (Hass et al., 1999) gives it a tighter bound of being in the NP class and time complexity $\mathcal{O}(\exp(cn^2))$ for a diagram $\mathcal{D}$ with n crossings, and some constant c, and space complexity $\mathcal{O}(n^2 \log(n))$. In terms of unknotting by Reidemeister moves, it has been shown that the number of moves required to reduce a diagram of an unknot is at most polynomial (Lackenby, 2015), hence one approach to determining the unknottedness of a diagram is by going through all possible Reidemeister moves sequences, however that would require an exponential time.*

It should also be noted, that the algorithm presented here doesn't guarantee that an unknot would be fully reduced, as the number of crossings is required to either lessen or remain the same in each loop pass, however there are known unknot diagrams which cannot be reduced without at some point increasing the number of crossings (Henrich and Kauffman, 2014). However, in the protein structures test set no such diagram was encountered. Similarly, the third Reidemeister move either wasn't used, or at most one segment-crossing pair was explored (and in significantly less than 1% of structures).

2.1.1. Finding a valid Dowker-Thistlethwaite code

As mentioned before, we can check if the projection of our structure on the plane (which corresponds to drawing a knot diagram) is a good representation of the topology, by checking the validity of its DT code (that is if all crossings are annotated by one odd and one even number). We do this by rotating the structure around its y-axis until the proper code is found (Equation 2.2), or the full rotation was completed – and then we change the coordinates of each bead $B_i = \{x, y, z\}$ to $B'_i = \{y, z, x\}$.

For each bead $B_i = \{x, y, z\}$ the coordinates are rotated around the y-axis by angle θ to get the new coordinates B'_i as follows:

$$\begin{aligned} x' &= x \cdot \cos\theta - z \cdot \sin\theta \\ z' &= z \cdot \cos\theta + x \cdot \sin\theta \\ B'_i &= \{x', y, z'\}. \end{aligned} \tag{2.2}$$

2.2. Smoothing the chain

Dowker-Thistlethwaite code reduction scheme described above could be applied to a projection of molecular backbone without any smoothing, but this would be computationally expensive and finding a valid (resulting in a proper code) projection, non-trivial. To simplify knot detection, the first step in the knot_pull workflow is the structure smoothing. This is done in a manner similar to the one described in (Taylor, 2000), with the topology conservation ensured by the triangle crossing condition introduced in (Koniaris and Muthukumar, 1991). Smoothing algorithm used (Algorithm 1) introduces additional breakpoints on the chain, which facilitates the collapse of the structure around the entanglements. This in turn both makes the termini of the chain protrude more, which makes finding a valid projection easier.

Algorithm 1 Smoothing algorithm for a sequence of 3D coordinates

INPUT: a doubly linked list of beads *atoms*
OUTPUT: a shorter doubly linked list of beads

function smooth (*atoms*)
while bead removed or moved by > 0.05Å **do**
  for each B_i in *atoms* **do**
    if less than two beads remain till end of chain **then**
      process B_{i+1}
    if no chain segment crosses $\triangle(B_i, B_{i+1}, B_{i+2})$ **then**
      if $[B_i, B_{i+2}] > 4$Å **then**
        move B_{i+1} to the $avg(B_i, B_{i+2})$
        process B_{i+1}
      else
        remove B_{i+1}
        process B_i
    else
      if $[B_i, B_{i+1}] > 4$Å **then**
        insert new bead $B_{i+1/2}$ at $avg(B_i, B_{i+2})$
        process $B_{i+1/2}$

Remark. *The time complexity of Algorithm 1 largely depends on the structure. Each smoothing step has a complexity of* $\mathcal{O}(n^2)$*, where* n *is the number of coordinates, however number of steps depends on a number of characteristics of the shape analysed, such as oblateness, and its ratio of length to the radius of gyration (how tightly packed is it).*

To the best of our knowledge there are no studies into the time complexity of structure smoothing algorithms on biological data. In our simulations, the number of smoothing steps required was influenced the most by:

- *proportion of coordinates not colinear with the line connecting the two termini;*
- *whether the main axis of the structure was roughly colinear with the line between two termini, or if it was curved.*

We run simulations on straight (Fig. 2.8 upper row) and curved (Fig. 2.8 lower row) helices of different lengths with with radius $A \in 1,..10$ units and slope K/A where $K \in 1/10..1/2+1...10$ units. Straight helices were generated with $n \in \{10, 120\}$ coordinates using the equations (with unit equal to 1):

$$\begin{aligned} \forall i \in \{1, ..., n\} \; x(i) &= A\cos(i) \\ y(i) &= A\sin(i) \\ z(i) &= Ki, \end{aligned} \tag{2.3}$$

for curved helices the y coordinate was changed to

$$y(i) = A\sin(i) + (i \text{ if } i < \frac{n}{2}, n - i \text{ otherwise })$$

Plots of the number of steps required for smoothing the initial n coordinates to just two points are shown in Figure 2.8. As can be seen there, for straight helices the number of steps in most cases either converges with length to a constant becomes regular in function of the slope. However, for curved helices, except in a small number of cases where the shape of a helix allowed for a very regular placement of beads, the number appears to constantly increase, which can be explained by the increasing deviation from the axis joining the termini.

Running times for a test set of real biological data (protein chains from the PDB database) are shown in Figure 2.9.

Additional advantage of collapsing the structure, instead of just simplifying it, is that it is centred on the entanglement. Which means, that for composite knots, each of the prime knots within, if they are separated on the chain, will collapse independently (example shown in Fig. 2.10). They can then be easily identified, by separating the structures into non-trivial fragments which cannot be simplified even when disconnected.

Algorithm 2 disconnects the continuous chain into two substructures in the middle of each chain segment separately. Each of the substructures is then smoothed separately. If this additional smoothing didn't change the coordinates of neither substructure, they are a potential decomposition of the structure into prime knots (or

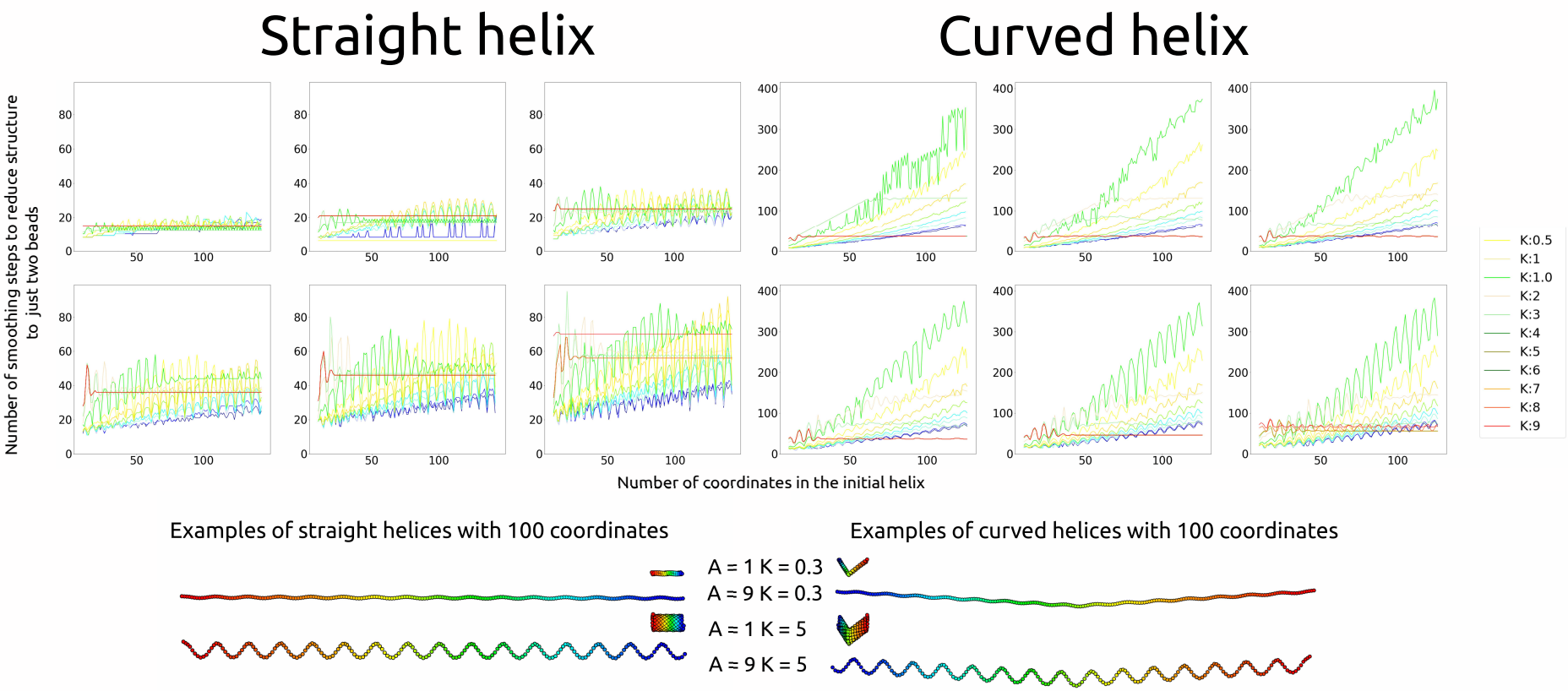

Figure 2.8: Number of smoothing steps required to reduce a helix to a two straight line connecting the two termini.

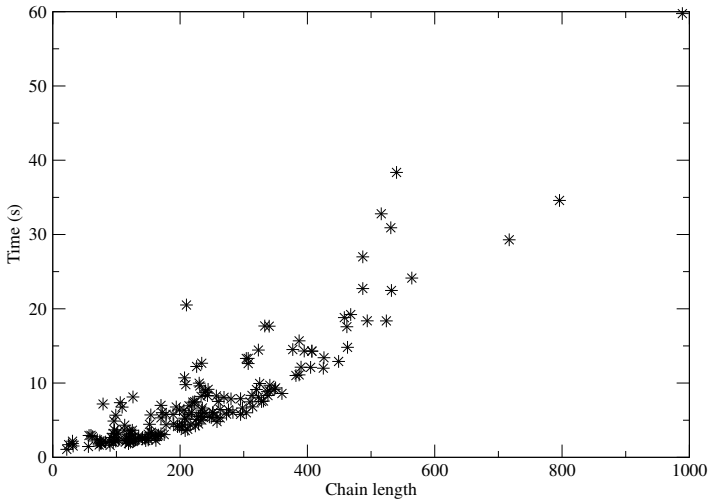

Figure 2.9: Running times for smoothing an unknotted protein chain by the length of the chain (a random subset of 50 000 protein chains from the RCSB PDB).

unknots). The algorithm for identifying linked chains in a structure is analogous, with the only difference being that the division into substructures is only done to separate the whole chains.

Algorithm 2 Detecting composite knots on a chain

INPUT: a doubly linked list of beads after smoothing *atoms*
OUTPUT: a list of disconnected sub-lists of beads *subchains*

edges is a list of beads preceding any edge that separates sub-entanglements
for each B_i in *atoms* **do**
  $midpoint = avg(B_i, B_{i+1})$
  if smooth(*atoms*) is the same as smooth(*atoms* to *midpoint*) + smooth(*atoms* from *midpoint*) **then**
    add B_i to *edges*
for each B_i in *atoms* **do**
  if $B_i...B_{i+k}$ for $k >= 4$ not in *edges* **then**
    add list $B_i...B_{i+k}$ to *subchains*
    process B_{i+k+1}
  else
    process B_{i+1}

2.2.1. Link detection

Knot_pull does not currently recognise different types of inter-chain links, but it does detect if there is a link (Fig. 2.11). Usually, links in proteins (in which each protein chain

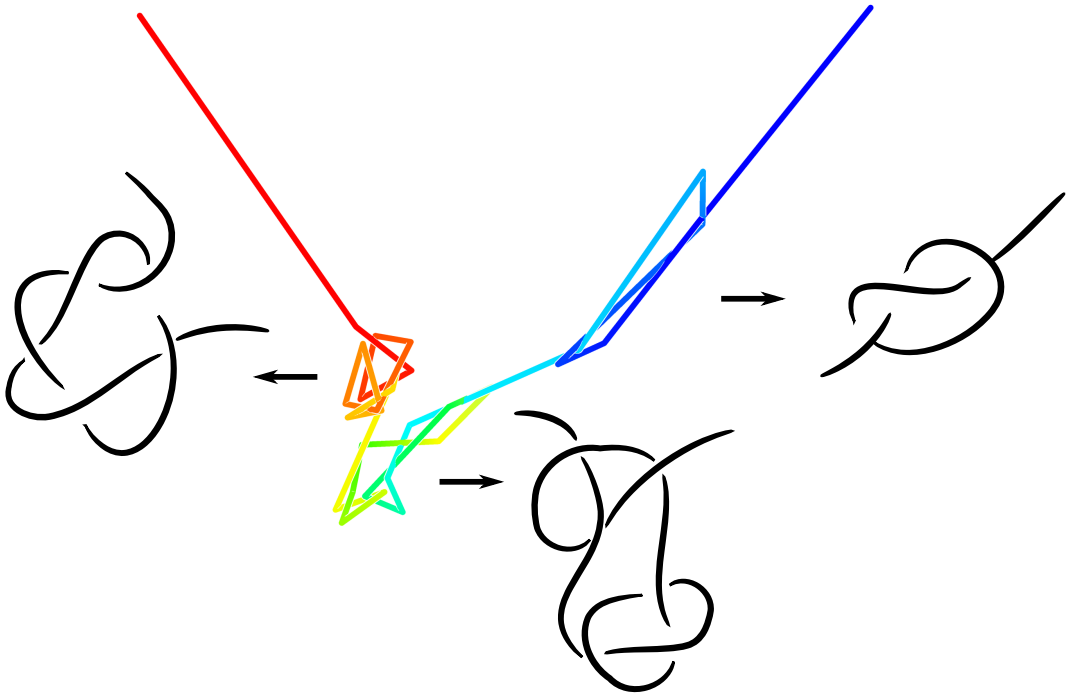

Figure 2.10: Collapsing the structure of a model of a mouse chromosome reveals three consecutive knot-like folds (from the N terminus – blue): 3_1, 5_1 and a composite $3_1\#3_1$.

is one link component) are detected in a manner which is an extension of the polynomial based knot detection – that is each chain is separately, and through random path closed, and the linking of so closed chains is calculated. This is an error-prone method, as the large loops used as closing paths (passing on a surface of an implicitly infinite sphere) are advantageous in reducing the number of crossings in single component knot diagrams, but counterproductive for links, as the probability of artificial linking between such loops increases. Knot_pull bypasses this limitation by again disregarding any closing – links are present when presence of one chain hampers the smoothing of another. This is checked in a manner similar to sub-entanglement detection. If after the smoothing of the whole structure finishes it is possible to further simplify the chains by considering them separately, they are linked. However, depending on the exact definition of a protein link (chains wrapped around each other vs. fragment of a chain passing through another), this approach can yield some false positives, as two crescent shaped structures interlocked would be marked as linked.

2.3. Topology preservation while smoothing

In any polymer smoothing algorithm, the crucial part is how to ensure that the topology will not be changed, that is, that the chain will not pass through itself. In knot_pull this is ensured by checking that no segment of the chain passes through the implicit

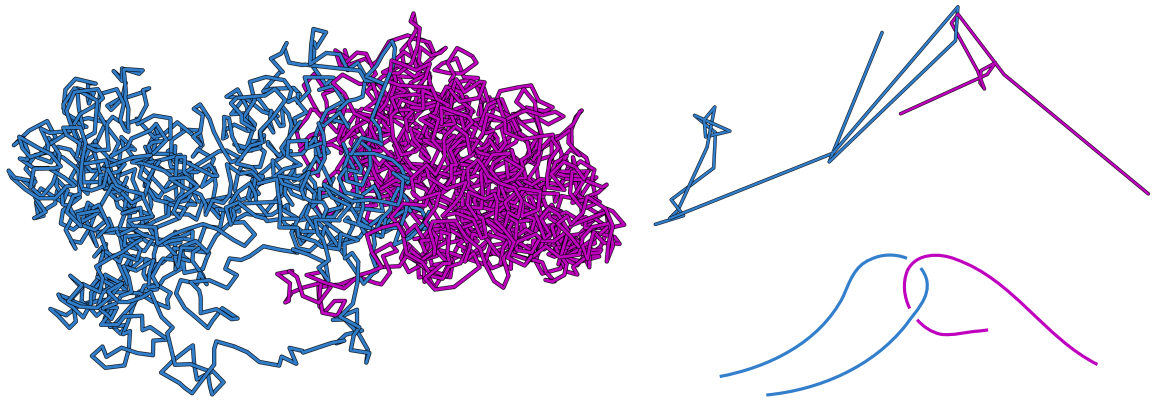

Figure 2.11: Chains found by knot_pull as linked.

path of the fragment that is smoothed (Figure 2.12).

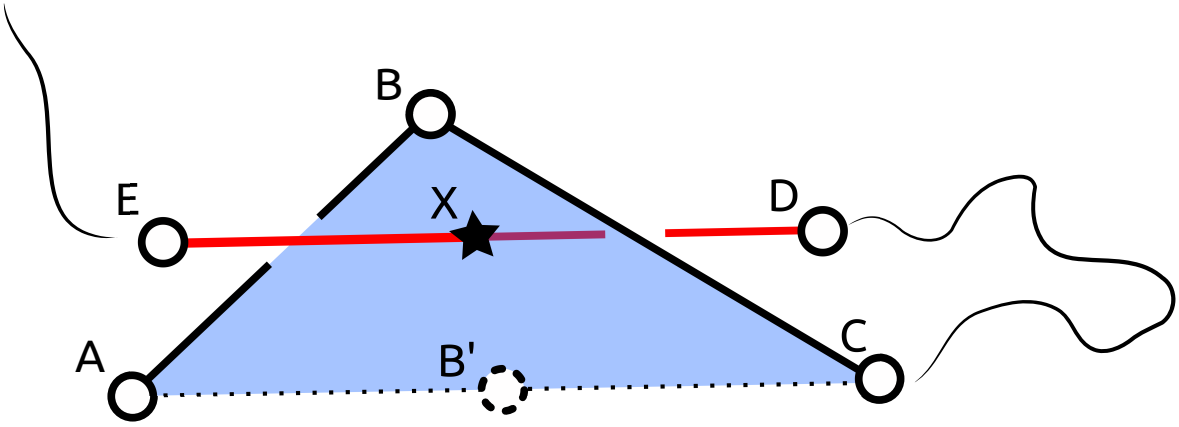

Figure 2.12: Obstacle for smoothing. Segment between beads D and E crosses through the surface spanning between beads A, B and C, thus preventing the smoothing algorithm from moving bead B to the coordinates B'.

For a chain made up of a sequence of N beads (3D coordinates) $B = \{B_1, B_2, ..., B_N\}$, the obstacles are found as follows: for a triangle $\triangle(B_i, B_j, B_k)$, where $i + 1 = j$ and $j + 1 = k$, all chain segments connecting beads $[B_a, B_b]$, where $a + 1 = b$ and $(b \leq i-1$ or $k+1 \leq a)$ are checked. First, we check if either B_a and B_b lay on the surface of the triangle by finding their projection on its surface (Heidrich, 2005), and calculating

its distance from the original point. For a point B_x, this is done as follows

$$\begin{aligned}
\vec{u} &= B_j - B_i \\
\vec{v} &= B_k - B_i \\
\vec{n} &= \vec{u} \times \vec{v} \\
\vec{w} &= B_x - B_i \\
\gamma &= [\vec{u} \times \vec{w} \cdot \vec{n}]/\vec{n}^2 \\
\beta &= [\vec{w} \times \vec{v} \cdot \vec{n}]/\vec{n}^2 \\
\alpha &= 1 - \gamma - \beta \\
B'_x &= \alpha B_i + \beta B_j + \gamma B_k,
\end{aligned} \tag{2.4}$$

where B'_x are the coordinates of a B_x projection on the plane spanned on the $\triangle(B_i, B_j, B_k)$. If $0 \leq \{\alpha, \beta, \gamma\} \leq 1$, B'_x lays within the triangle. Now, if $B_x = B'_x$ (or $|B_x - B'_x| < \epsilon$ to correct for a possible numerical error), B_x lays within the triangle and is an obstacle in smoothing.

If no obstacle was found yet, we check the segment $[B_a, B_b]$ by finding the intersection between this segment and the plane on which lays $\triangle(B_i, B_j, B_k)$. Normal is the cross product of two edges of the triangle:

$$\begin{aligned}
\vec{u} &= B_j - B_i \\
\vec{v} &= B_k - B_i \\
\vec{n} &= \vec{u} \times \vec{v},
\end{aligned} \tag{2.5}$$

from which we can get a 4D vector of the plane $\vec{N} = \{a, b, c, d\}$, where $\vec{n} = \{a, b, c\}$ and

$$d = \sum \{(\vec{n} \cdot B_i)\} \cdot -1.$$

With $\vec{x} = B_b - B_a$ corresponding to the checked segment, we check if the plane and the line on which lays $\vec{x}$ intersect (are not parallel): if

$$\vec{n} \cdot x \neq 0,$$

the point of intersection exists, and we can calculate it as:

$$\begin{aligned}
\vec{p} &= \vec{n} \cdot \frac{d}{\vec{N} \cdot \vec{N}} \\
\vec{w} &= B_a - \vec{p} \\
pos &= -1 \cdot \frac{\vec{n} \cdot \vec{w}}{\vec{n} \cdot x} \\
\vec{u'} &= \vec{u} \cdot pos \\
\vec{I} &= B_a + \vec{u},
\end{aligned} \tag{2.6}$$

where $\vec{I}$ is the intersection between the line and the plane, and *pos* indicates where on the vector $[B_a, B_b]$ it lays – it belongs to the segment only if $0 \leq pos \leq 1$. Now we can check if point $\vec{I}$ lays within the triangle using Equation 2.4.

2.4. Validation of results

Knot_pull was tested against the KnotProt database (Jamroz *et al.*, 2014), which is hand curated. All the chains annotated in the database as knotted (1054 in total) have been found to return the same knot type for the full chain (disregarding the chirality, as it is not returned by knot_pull). For 503 slipknotted chains there were 3 cases where knot_pull reported a knot instead of the (expected) unknot – all three structures actually contain an (albeit very shallow) knot. Additionally, over 94% of those structures were reduced to only 2 points during smoothing. From chains reported by KnotProt as trivial, 65000 were randomly selected and tested, and all of them reduced to 2 points after smoothing, except for seven structures which actually contain knots (Table 2.1), and one which by homology should be knotted, but was calculated as an unknot due to a gap in the structure (PDB Id 4kjs chain A). Testing available RNA structures confirmed results of (Micheletti *et al.*, 2015), with only the structures reported there exhibiting non-trivial topology.

Table 2.1: Protein chain which were found by knot_pull to contain different topology than reported by KnotProt

PDB Id	Chain	Knot type (KnotProt)	Knot type (knot_pull)	Note
5nfj	A	0_1 (unknot)	3_1	
5ush	A	0_1 (unknot)	3_1	
4r70	A	0_1 (unknot)	4_1	
5wr7	A	0_1 (unknot)	3_1	
5l6t	A	0_1 (unknot)	3_1	
5hya	A	0_1 (unknot)	3_1	
4coq	A	0_1 (unknot)	3_1	very shallow knot
1oq5	A	0_1 (slipknot: 3_1)	3_1	
535u	A	0_1 (slipknot: 3_1)	3_1	
3f7u	A	0_1 (slipknot: 3_1)	3_1	

One obvious weakness of all available biomolecular knot detection software, from which knot_pull is not exempt, is its uncertainty when faced with structures with missing residues or otherwise gapped backbone. Such gaps are filled by straight lines

which can often result in artificially knotted (or unknotted) structures. Chapter 4 presents GapRepairer – a server designed to provide the gap filling step in analyzing protein structures.

It has frequently been suggested that proteins may not be pure chemical entities but may consist of mixtures of closely related substances with no absolute unique structure. The chemical results obtained so far suggest that this is not the case, and that a protein is really a single chemical substance, each molecule of one protein being identical to every other molecule of the same pure protein.

Frederic Sanger 1952 *The arrangement of amino acids in proteins.* Adv. Protein Chem

3

Modeling protein sequence evolution

MOLECULAR evolution is hard to trace – we don't know the intermediate steps. While when studying evolution on macroscopic scale it is possible to encounter some fossils giving us an insight into the transitional forms, there is no record of sequence mutations that may have happened along the way. The only information available are the sequences found in (usually only) currently living organisms, and then only a subset of them. A popular approach in phylogenetics is parsimony – a phylogenetic Occam's razor, where the most likely explanation is the simplest one – according to which the correct phylogeny is the one explainable with the least number of evolutionary events. While based on this principle, it is possible to create a possible model of evolution of related sequences, there is no way of it is the correct one. This model may underestimate the number of mutations, or if the sequences variations not found on a tree are impossible (due to lethality of mutations), lost due to e.g. extinction or simply not yet discovered.

This chapter introduces an as-of-yet unpublished algorithm for the multiple alignment of sequence profiles (Sec. 3.1), and describes results published in (Jarmolinska *et al.*, 2019b) (Sec. 3.3.2), (Lamb *et al.*, 2019) (Sec. 3.3.3) and (Dabrowski-Tumanski *et al.*, 2015) (Sec. 3.3.4).

3.1. Maximum weight trace finding – an algorithm for multiple alignment

Multiple sequence alignment (MSAs) are generally made with the assumption of a common evolutionary lineage of studied sequences. However, a common lineage does not necessarily indicate a noticeable similarity between all pairs of sequences in our data set. As such MSAs are built iteratively, often using a rough similarity tree as a guide, in each step merging the two sub-alignments (at the beginning – single sequences) that are most similar to each other. An important caveat here, is that the alignment process is trying to maximise the overall score of matching. Local alignments within a global one may be significantly suboptimal.

An alternative method of building an MSA is by maximising the agreement with all-vs-all pairwise (which can be resolved optimally in polynomial time – Section 1.1.2) alignments. This can be done by finding a maximum weight trace (Kececioglu, 1993) in a graph $\mathcal{G} = (V, E, \prec)$ representing a set of alignments, which is defined as follows.

Definition 2. *A graph $\mathcal{G} = (V, E, \prec)$ is an alignment graph for a set $\mathcal{S}$ of sequences if vertices V correspond to positions in sequences in S, with an order within each sequence S_i imposed by relation $\prec$ on positions $s_i, s_j \in S_i$: $s_i \prec s_j \iff i + 1 = j$, that is $\prec$ holds only if s_i immediately preceeds s_j. Edges E are undirected weighted connections between vertices which have been aligned in one of the pairwise alignments. Position s_i is characterised by two values: sequence from which it was taken $s_i.sequence$, and its index in this sequence $s_i.index$.*

A path in graph $\mathcal{G}$ is a set of positions to be aligned in one column – thus splitting the graph into connected components gives columns of the alignment, with the caveat that the alignment is only valid if columns can be linearly ordered under relation $\prec'$, where for connected components A and B:

$$A \prec' B \iff (\exists a \in A)(\exists b \in B) : x \prec y.$$

A trace in the alignment graph $\mathcal{G}$ is then a subset of edges $T \subseteq E$ for which connected components are acyclic under $\prec'$. For a graph $\mathcal{G}$ with edges weighted by function w the maximum weight trace is found by maximising $\sum_{e \in T} w(e)$.

The problem of finding the maximum weight trace of an alignment graph has been shown to be NP-complete (Kececioglu, 1993). The main obstacle for biological data is that the pairwise alignments may disagree – leading to the cycles between connected components. Here we present two heuristics based on greedy graph traversing with a modified Dijkstra's algorithm (Dijkstra, 1959) for the shortest-path tree. We assume

a data set of M sequences of length N, which positions all have at least one match to another position (a connected edge in the graph), with the weight w of edge describing its length (thus we will be *minimising* the weight trace).

Note. The requirement that all positions in sequences have been matched to something doesn't necessarily hold in real data – depending on the input those vertices can be first filtered out (and the relation $\prec$ modified to mean "immediately preceeds amongst values present in the graph"), or as trivial connected components treated as ready-made columns.

3.1.1. Depth-first column building

In this approach the set of columns $\mathcal{C}$ is build sequentially, by column. We start by finding the shortest edge $e(v_i, v_j)$, and marking the vertices it connects as belonging to the new column C. Then we find the shortest edge going out of the column (connecting a vertex in column with some vertex v_k outside), check if $\not(\exists v_i \in C) v_i.sequence = v_k.sequence$, and if not we add v_k to C. When no outgoing edges are available the column is ready, and added to $\mathcal{C}$. Each edge is removed when first encountered.

During the construction of next columns, we additionally check if adding v_k to C would create cycles in the ordering of columns, that is adding v_k would cause

$$(\exists c_i \in \mathcal{C})\ c_i \prec' C \wedge C \prec' c_i.$$

It should be noted, that as the algorithm is only edge-based, all vertices without any outgoing edges will be discarded – to keep them in the alignment they should be later added as one-character columns.

Lemma 2. *Time complexity of Algorithm 3 on a graph $G = (V, E)$ is $\mathcal{O}(|E|^2 \cdot (|E| \cdot |V|^2))$.*

Proof. Three loops can run $\mathcal{O}(E \cdot E)$ times, with the worst case scenario being that for each edge removed from E in the outer while loop, we will pass through all the remaining edges in the for loop without removing anything. Second term is caused by the cycle search – for each edge found for a column we need to compare each vertex in this column (at most $|V|$ elements) against all other columns (totalling up to V elements). ■

3.1.2. Breadth-first column building

This approach is based on a bottom-up clustering of graph vertices into columns (Algorithm 4). At the beginning, each column consists of one vertex. In each step we

Algorithm 3 Heuristic for finding a shortest length trace in an alignment graph by sequentially extracting columns

INPUT: list of edges E sorted by lowest length
OUTPUT: ordered list of columns C
C is an ordered list of columns
while E **do**
    $e(v_i, v_j) \leftarrow E.pop\_first()$
    $c = [v_i, v_j]$ list representing a new column
    while *change* **do**
        $change = 0$
        for $e(v_a, v_b) \in E$ **do**
            if $v_a \in c \wedge v_b \in c$ **then**
                remove e from E
            else
                if $v_a \in c$ XOR $v_b \in c$ **then**
                    remove e from E
                    if $v_a \in c$ **then**
                        if $v_a.sequence \notin c \wedge c + v_a$ will not add cycles to C **then**
                            add v_a to c
                            $change = 1$
                            break
                    else
                        if $v_b.sequence \notin c \wedge c + v_b$ will not add cycles to C **then**
                            add v_b to c
                            $change = 1$
                            break
    if c will not add cycle to C **then**
        add c to C

find the shortest edge $e(v_i, v_j)$ where $v_i \in C_i \wedge v_j \in C_j \wedge C_i \neq C_j$. Then we verify if sequences present in C_i, C_j do not overlap, and column $C_k = C_i + C_j$ would not introduce cycles in the ordering of columns. If there is no such problems, we merge columns. Regardless of merging, the edge is removed. Algorithm finishes, when there are no more edges connecting different columns.

Algorithm 4 Heuristic for finding a shortest length trace in an alignment graph by bottom up clustering of columns

INPUT: PriorityQueue of columns C, where each column c is a PriorityQueue of vertices v sorted by shortest edge out of column available, with one vertex each
OUTPUT: unsorted list of columns C_L

while *change* $\wedge$ C **do**
  column $c \leftarrow C.pop\_first()$
  column $v \leftarrow c.pop\_first()$
  if v has an edge $e(v, v')$ where $v' \notin c$ **then**
    $c' \leftarrow v'.column$
    remove c' from C
    put v in c
    if $v.sequence \notin c' \wedge v'.sequence \notin c \wedge c + c'$ will not add cycles to $C \wedge c +$ c' will not add cycles to C_L **then**
      $c = c + c'$
      remove internal edges in c
    else
      remove e from v and v'
      put c' in C
    put c in C
  else
    add c to C_L

Lemma 3. *Time complexity of Algorithm 4 on a graph $G = (V, E)$ is $\mathcal{O}(|E| \cdot |V|^2)$.*

Proof. While loop in each step either removes one edge from E or removes one column from C (which is equal to $|V|$ at the beginning, so $|E| + |C|$ is the upper bound on the number of passes). Whenever an edge is removed we check if honouring it will add cycles to the ordering of columns, which needs up to $|V|^2$ comparisons (up to $|V|$ elements in $c + c'$ against all the other vertices in other columns in C). Here we disregard the costs of removing elements from priority queues. ■

3.1.3. Results

Due to the specific nature of biological data the depth-first heuristic appears to run about faster 20% faster. Alignments produced by both methods are shown in Fig. 3.1.

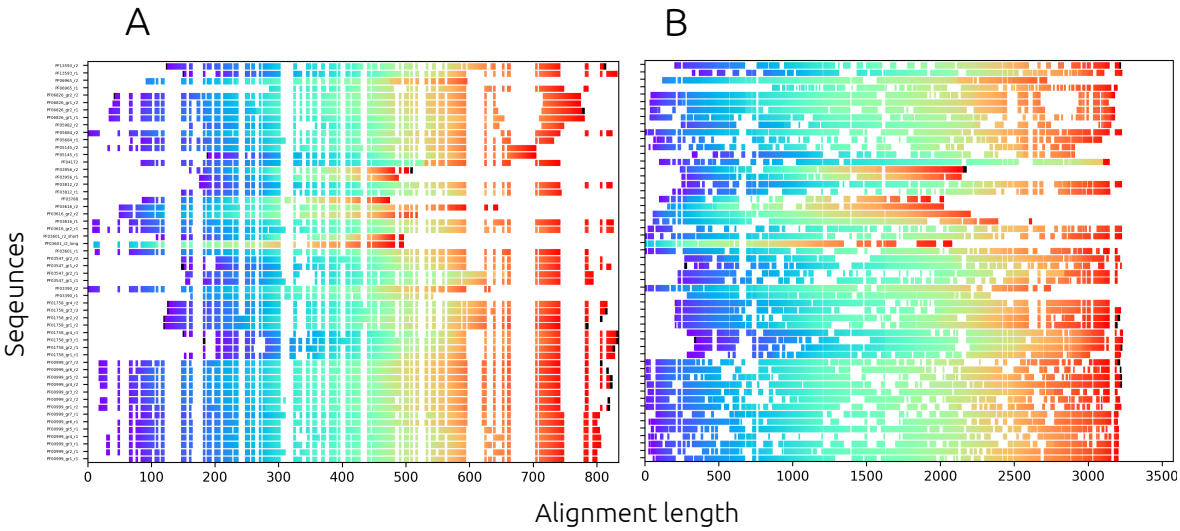

Figure 3.1: Exemplary multiple profile alignment for data described in Section 3.2 created by (A) Algorithm 3 and (B) Algorithm 4. Each row corresponds to one sequence, white spaces are gaps in the alignment, positions are coloured according to a rainbow colour scale spanning the whole length of the initial sequence. Terminal residues of each sequence are coloured blak when present. Blue ovals indicate discontinuities in sequences (indicated by a break in the colour scale).

When compared with multiple sequence alignment methods, the optimised weight trace heuristics shown are more suited to a local alignment. In particular in depth-first column building only the parts of sequences which were matched to something in initial pairwise alignments have a chance to be included. Additionally, if a given sequence has no reasonable similarity (significance below the default cut-off) to any other, it will not be included in the final alignment – on the contrary to the usual behaviour of MSA software, where all the initial sequences are mandatory. It is important to note, that the resulting alignment, due to its local nature, does not guarantee that the fragment present in the alignment will contain all the positions in its range (Fig. 3.1).

One important advantage that this approach has on more common score-maximising MSA algorithms, is that the *sequences* aligned need only be pairwise-alignable and have the $\prec$ relation defined – they don't need to be scored in triplets or larger set, which is a requirement in other methods. This makes trace methods uniquely suited to the alignment of sequence profiles, with one example of such analysis described in the next section.

Python code implementing the full aligning workflow is available online at `http://github.com/dzarmola/hhaligner`.

3.2. Case study for the multiple profile alignment algorithm – evolution of repeated membrane proteins

Membrane proteins are an interesting case in protein evolution, as many of them exhibit internal symmetry suggestive of domain duplication and fusion (Duran and Meiler, 2013). In some proteins the fusion of chains of a dimer into a single molecule is credited with the emergence of a knotted topology. However, an even more interesting example can be found amongst the cation:proton antiporter/anion transporter (CPA_AT) protein clan of membrane transporters and related families. Based on sequential and structural similarity it is possible to find 16 protein families which appear to be paralogous[1] – we have found organisms with representatives of up to 13 of those families.

Most of families in the data set have two instances of a common structural motif, to which we will refer to henceforth as a "repeat":

- 12 families belonging to the CPA_AT clan (CL00064 in Protein Families (PFam) database (El-Gebali *et al.*, 2018));
- transition state regulatory proteins (AbrB) (PF05145 in PFam);
- 2-hydroxycarboxylate transporters (2HCT) (PF03390 in PFam);

Additionally, two families with sequences covering only a "single repeat" were included – LrgA and LrgB. Presence of both single domain and repeated proteins makes this an already interesting topic, but this is additionally spiced up by the topological angle – all the repeated proteins from this data set for which the structure is known are trivial, with the exception of those belonging to the 2HCT family, which are slipknotted. Co-existence of single and double domain proteins, as well as topologically different repeated structures suggested that the common ancestor of all studied proteins was a single domain protein, similar to LrgA and LrgB. Thus, to trace all the evolutionary events which resulted is such a diverse group of descendants we have decided to "break" all the duplicated proteins, and create a phylogenetic tree of all the resulting single repeats.

Due to the overall diversity of repeats found in the data set we have first clustered them by the sequence similarity, which resulted in further division of the largest families into smaller groups. Overall sequence similarity of the data set was not enough to create

[1]Paralogue is a homologue – protein with the same ancestor – within the same organisms, arising from a gene duplication.

a workable alignment for phylogenetic analysis, thus we used the algorithm described in the Section 3.1, to align instead profiles calculated with HH-Suite (Steinegger *et al.*, 2019) for each family-group-repeat, the resulting alignment is shown in Fig. 3.1 (B). We used this alignment, as well as additional information provided by sequence clustering to calculate a bayesian inference tree using MrBayes (Huelsenbeck and Ronquist, 2001). Fig. 3.2 shows a schematic unrooted tree showing the evolution of studied proteins.

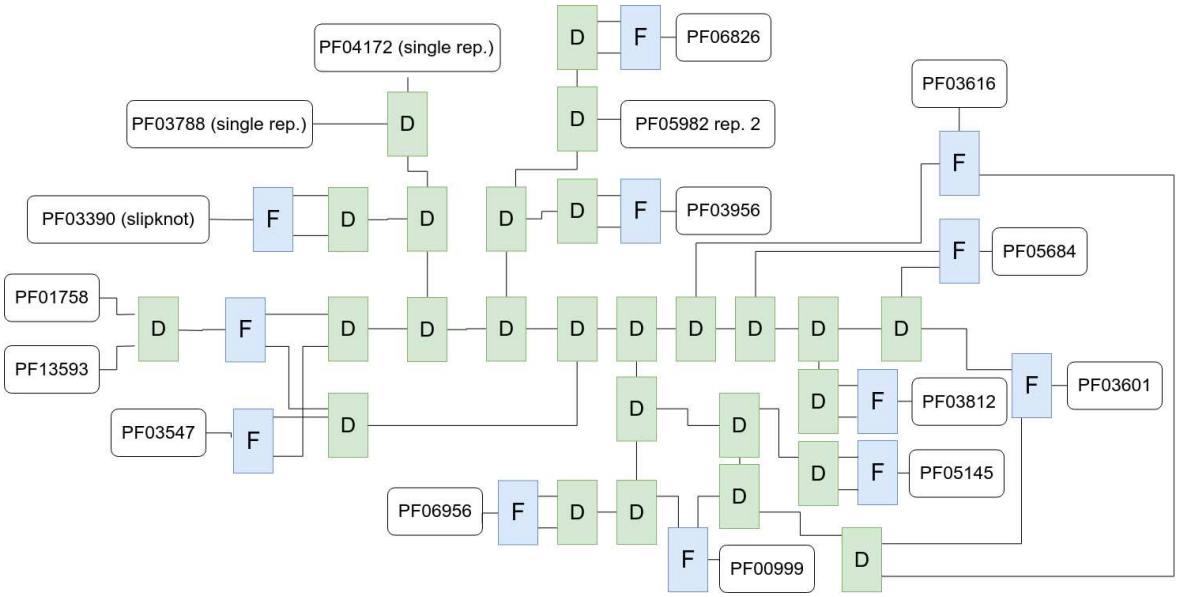

Figure 3.2: Schematic unrooted tree of evolution of 16 families of membrane proteins. Tree is annotated with PFam family identifiers. Green boxes indicate gene duplications, blue indicate gene fusions.

Our results suggest that topological difference of studied proteins is the result of separate fusion events. We expect there to be 13 different fusion events between repeats from the data set – there are two not repeated families, one of the families from the CPA_AT clan appears to have duplicated and speciated after fusion (PF13953 speciated from PF01758), and one of the repeats (sequentially first from family PF05982) didn't shown sufficient similarity to any other repeat to be included in the alignment, and can in fact be of different ancestry. The most interesting sequence of events appears to have happened with families PF01758 and PF03547 – as they share very similar repeats, but in reverse order in the sequence.

3.3. Co-evolution-based structure analysis

Protein structures are the basis of their function – an active site of an enzyme needs specific residues to properly interact with a ligand. Positions of those residues are often conserved, to a point where they can be used to predict some interaction based on sequence patterns. It's a prime example of how the function (for which proper binding of the ligand is crucial) and structure (shape of the binding site which need particular

residue properties in specific places) influence the evolution of the sequence. Obviously, the function (or lack thereof) of a folded protein cannot directly exert evolutionary pressure on the sequence. However, any mutation in the DNA which impairs either the folding or functioning of the resulting protein, impairs in turn all the processes which depend on this molecule, possibly leading to the death of the cell – which as a result removes this mutation from the evolutionary lineage.

One way through which significant (thus potentially lethal) changes can be conserved throughout evolution is through compensating changes – the effects of one change can be mitigated through a different mutation (Taylor and Hatrick, 1994). By looking at this process in reverse – if two positions in a sequence appear to be correlated, we can assume that they are facing a joint evolutionary pressure, which is usually the case for residues close together in the 3D structure. One of the methods of studying such dependencies is through Direct Coupling Analysis (DCA). For a multiple sequence alignment, DCA calculates a fully-connected statistical model of residue probabilities, which gives direct (not through another residue) correlation scores, called Direct Information (DI), to all pairs of positions in the MSA.

3.3.1. Direct Coupling Analysis

Direct Coupling Analysis (DCA) is a statistical inference framework built on a Potts model, which represents interacting spins (with q possible values) on a lattice model (a generalized Ising model). Standard Potts model has an interaction Hamiltonian given by:

$$H_p = -J_p \sum_{(i,j)} \delta(s_i, s_j), \tag{3.1}$$

where J_p is the interaction parameter, δ is the Kronecker delta (1 if s_{ij}, 0 otherwise), $s_i \in 1...q$ is the spin at position i, and (i, j) are pairs of positions which are considered for interaction (e.g. nearest neighbours).

Potts model can be generalised by adding a "magnetic field" term, and allowing the parameters to vary across model, which for a fully connected model gives:

$$H_p = -\beta \sum_i \sum_j J_{ij} \delta(s_i, s_j) - \sum_i h_i s_i, \tag{3.2}$$

in which

$$\beta = \frac{1}{kT},$$

where k is the Boltzmann constant and T the temperature.

To model sequence evolution this model can be calculated to fit the multiple sequence alignment of, expected to be phylogenetically related, sequences. For an MSA

of length N the model would be an $N \times N$ lattice with $q = 21$ spins (representing the symbols in the alignment – 20 standard amino acids and a gap). For the alignment as a whole, each spin on the lattice is in fact better presented as a $q \times q$ matrix of Kronecker deltas of coincidences between corresponding columns in the alignment. Under this model we can calculate the probability of a sequence $a = a_1, a_2...a_N$ to match the model, as

$$P(a|J,h) = \frac{1}{Z} \exp(\sum_{i=1}^{N-1} \sum_{j=1}^{N} J_{ij}(a_i, a_j) + \sum_{i=1}^{N} h_i(a_i)), \tag{3.3}$$

where Z is the normalisation constant, so that $\sum_a P(a|J,h) = 1$, $h_i(a_i)$ is a value representing the propensity of a symbol (e.g. amino acid) to be found at i-th position, $J_{ij}(a_i, a_j)$ is a value representing the propensity of a symbol a_i to be found at i-th position, and a_j to be found at j-th position in the same sequence.

DCA seeks a distribution P which maximises the functional

$$\begin{aligned} F[P] = &- \sum_a P(a) \log P(a) \\ &+ \sum_{i<j} \sum_{x,y} J_{ij}(x,y)(P_{ij}(x,y) - f_{ij}(x,y)) \\ &+ \sum_i \sum_x h_i(x)(P_i(x) - f_i(x)) + Z(1 - \sum_a P(a)), \end{aligned} \tag{3.4}$$

where f_i are observed frequencies of symbols at position i, f_{ij} is the covariance of symbols positions i, j derived so that it will reproduce the observed marginal distributions, Z is the normalisation constant, and h_i and J_{ij} are the model parameters. The parameters are optimised to satisfy the empirical (from the alignment) frequencies and correlations. It is important to note, that the model is intractable on a multiple sequence alignment, as the normalisation constant Z would require a number of terms equal to the number of all possible sequences of a given length – N^q.

There are multiple derivations of the DCA model, all leading to the estimation of model parameters maximising the likelihood of the MSA:

- mpDCA (Weigt *et al.*, 2009): inference based on message passing/belief propagation;
- mfDCA (Morcos *et al.*, 2011): inference based on a mean-field approximation;
- gaussDCA (Baldassi *et al.*, 2014): inference based on a Gaussian approximation;
- plmDCA (Ekeberg *et al.*, 2013): inference based on pseudo-likelihoods.

Strength of interaction, also called Direct Information (DI), between two columns i, j in the alignment can be calculated using a Frobenius norm and an average product correction, as (Ekeberg *et al.*, 2013; Baldassi *et al.*, 2014):

$$DI_{ij} = F_{ij} - \frac{F_i \cdot F_j}{F}, \tag{3.5}$$

where

$$\begin{aligned} F_{ij} &= \sqrt{\sum_{a,b} J_{ij}(a,b)^2} \\ F_i &= \frac{1}{N} \sum_{i \neq j}^{N} F_{ij} \\ F &= \frac{1}{N^2 - N} \sum_{i,j,i \neq j} F_{ij} \end{aligned} \tag{3.6}$$

and a, b are spins (symbols) found in the alignment.

Direct Information indicates the most co-dependent pairs of sequence positions, which implies the importance of their cooperation for the proper functioning of the molecule. Such pairs can give new insights into various forms of structural interactions, which makes co-evolutionary measures useful for a broad range of applications, such as structure prediction (Sułkowska *et al.*, 2012a; De Leonardis *et al.*, 2015; Michel *et al.*, 2017; Taylor and Sadowski, 2011), conformational dynamics (Morcos *et al.*, 2013), analysis of folding pathways (Dabrowski-Tumanski *et al.*, 2015; Cheng *et al.*, 2014; Jamroz *et al.*, 2014) and prediction of interaction partners and interface (Dos Santos *et al.*, 2015; Schug *et al.*, 2009; Cheng *et al.*, 2014; Zschiedrich *et al.*, 2016; Skwark *et al.*, 2017; Bitbol *et al.*, 2016)

3.3.2. DCA-MOL – mapping co-evolution to a structure

DCA-MOL is a plugin we created for PyMOL Molecular Graphics Systems which facilitates the analysis of Direct Coupling Analysis results. One of the hurdles to cross when analysing co-evolutionary scores is that DCA methods is that the Direct Information scores are given for all possible position pairs, with little indication of which are significantly correlated. DCA results are highly dependent on the input data, such as the number and diversity of the sequences, and depending on the desired application can gain significantly from some additional knowledge (e.g. when studying the intricacies of a folding pathway, knowing the final structure can help differentiate between folding and native interactions). DCA-MOL performs automatic mapping between the structure, sequences in the MSA and DI results, taking into account any missing or discarded residues (using a modified Needleman-Wunsch algorithm –

Alg. 5), and provides the user with a convenient and fully interactive plot coupled with the structure visualisation.

Lemma 4. *Time complexity of Algorithm 5 is* $\mathcal{O}(N \cdot M)$*, space complexity is* $\mathcal{O}(N \cdot M)$*.*

Proof. First part of the alignment creates two matrices of size $N \times M$ in $N \times M$ steps. Second part backtracks through the matrix in at most $N + M$ steps. ■

Case study 1 – Interface interactions in isocitrate dehydrogenase protein dimers.

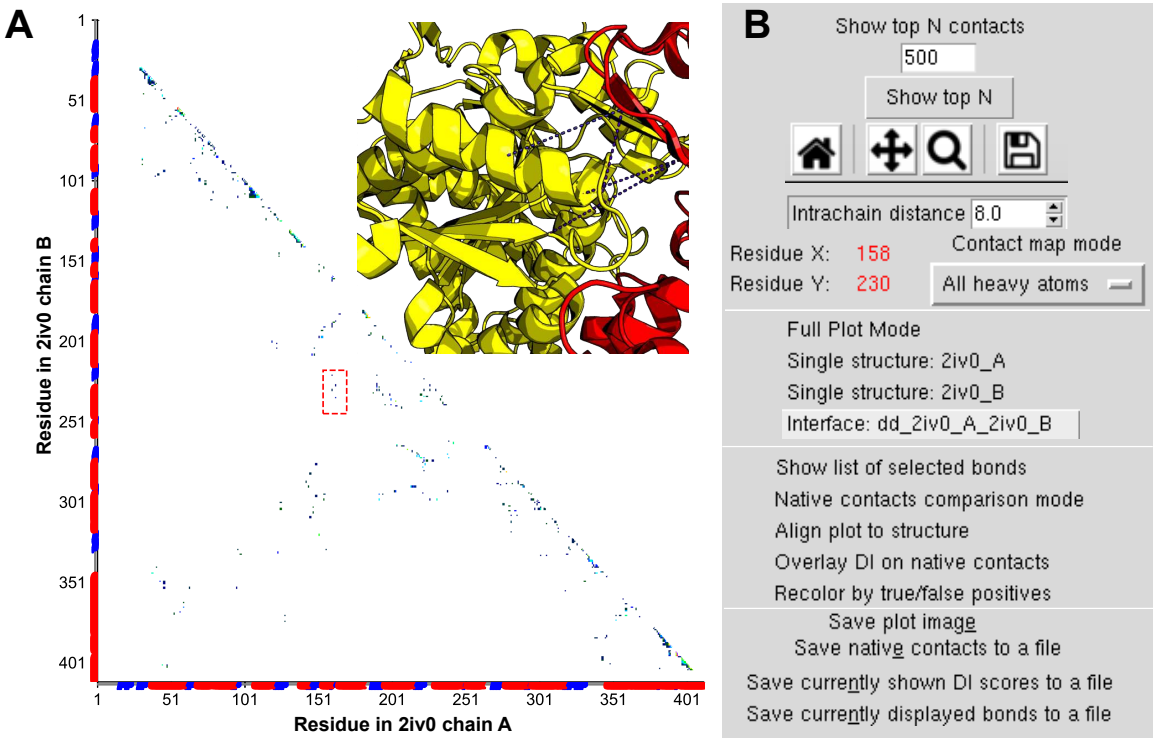

Figure 3.3: DCA-MOL analysis of isocitrate dehydrogenase interface interactions (PDB Id 2iv0). (A) A DI map is shown in the lower triangle (x-axis for residues in chain A, y-axis for residues in chain B), and 3D structure in the upper triangle (red for chain A, yellow for chain B). The interactions between two chains are marked with a red rectangle in the DI map and black dash lines in the structure. (B) A collection of DCA-MOL's options and features. DCA output files used here were calculated using the DCA server (`http://dca.rice.edu`).

Interactions between residues in molecular interfaces could also be coevolved. DCA studies can not only be used to capture the interactions inside the protein monomer, but also be used to infer interfacial interactions in homo or heterodimeric systems. With DCA-MOL, users can switch and compare these interactions more efficiently.

In this sample case, we use isocitrate dehydrogenase dimers to showcase the molecular interface analysis (Fig. 3.3). Isocitrate dehydrogenase is a homodimer containing two chains (A and B) (PDB Id 2iv0 and Pfam Id PF00089.25). By including both

Algorithm 5 Modified Needleman-Wunsch algorithm for global alignment of almost identical sequences with additional gap information

INPUT: *Sequence1* of length N with symbols indicating structure breaks, *Sequence2* with length M

OUTPUT: aligned sequences *Aligned*1, *Aligned*2

M is an empty matrix of size $N \times M$, $M[0][0] = 0$, $Path$ is an empty matrix of size $N \times M$

for i=0 to N **do**
    $M[i][0] = M[i-1][0] - 1$
    $Path[i][0] = 1$

for j=0 to M **do**
    $M[0][j] = M[0][j-1] - 1$
    $Path[0][j] = 2$

for i=1 to N **do**
    for j=1 to M **do**
        if $Sequence1[i] \neq$ **then**
            Match = $M[i-j][j-1] + (2$ if $Sequence1[i] = Sequence2[j]$ else $-2)$
            Insert = $M[i][j-1] - 1$
            Delete = $M[i-1][j] - 1$
        else
            Match = $M[i-j][j-1]$
            Insert = $M[i][j-1] + 1$
            Delete = $M[i-1][j] - 1$
        $M[i][j] = max(Match, Insert, Delete)$
        if $Match = M[j][j]$ **then**
            $Path[i][j] = 3$
        else
            if $Insert = M[j][j]$ **then**
                $Path[i][j] = 1$
            else
                $Path[i][j] = 2$

```
Aligned1 = "", Aligned2 = ""
while i > 0 or j > 0 do
    if i > 0 and j > 0 and M[i][j] = Match then
        Aligned1 = Sequence1[i] + Aligned1
        Aligned2 = Sequence2[j] + Aligned2
        i = i − 1, j = j − 1
    else
        if i > 0 and M[i][j] = Insert then
            Aligned1 = Sequence1[i] + Aligned1
            Aligned2 = "-" + Aligned2
            i = i − 1
        else
            Aligned1 = "-" + Aligned1
            Aligned2 = Sequence2[j] + Aligned2
            j = j − 1
```

chains independently in the alignment file, loading a corresponding chain from the structure for each sequence, and indicating that they form a molecular interface, we are able to study them separately, and also as a dimer.

This mode allows switching and comparing between monomeric interactions and interfacial interactions efficiently by choosing different plot options in the drop-down list. In this example, we found that some top DI value pairs are far away from each other in the monomeric structure but are close in the dimeric structure. This observation suggests that these interactions could be important to keep quaternary structure and function.

Case study 2 – Conformational dynamics in periplasmic binding proteins.

While performing their function, some proteins experience large conformational changes. DCA-MOL can be utilised to study how co-evolving pairs play a role in different protein conformations. Here, we use DCA-MOL to illustrate the conformational change of L-leucine binding protein, [PDB Ids 1usg (open state) and 1usi (close state)] upon ligand binding.

Within the multimodel mode, we can switch and compare between different contact maps for different states (see *change current state* option). The native contact map (lower triangle) shows a similar pattern between two states, except for a set of contacts exclusive to the closed state (Fig. 3.4). In our predicted contact map (upper triangle), we do not only get the interaction information for single states, but also interactions

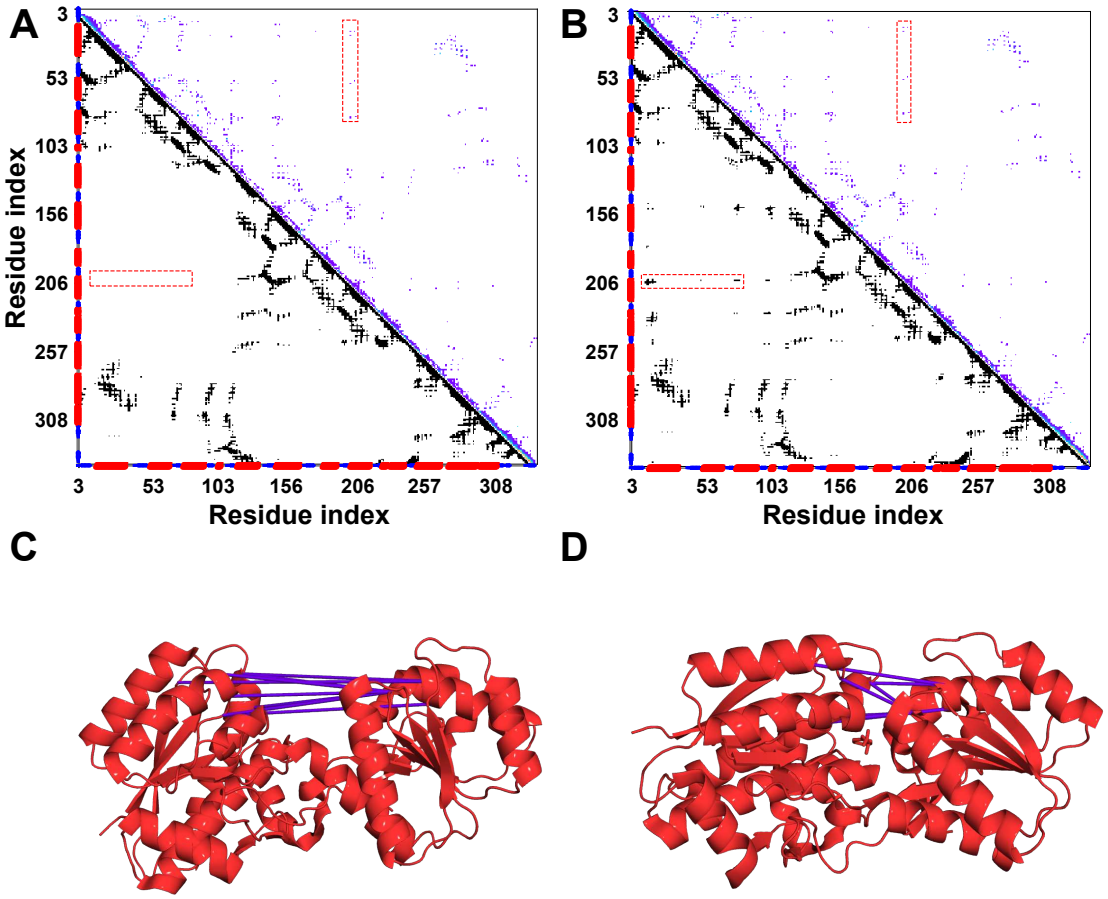

Figure 3.4: DCA-MOL analysis of L-leucine-binding protein: (A,C) Apo state (PDB Id 1usg); (B,D) closed, holo state (PDB Id 1usi). DCA-MOL's interactive plots of Direct Information (upper triangle) and contact maps of the structures (lower triangle). Selected with a red rectangle are contacts present only in the closed conformation (top). Cartoon representation of the structure (red). Predicted interactions selected on the plot are shown as purple bonds (bottom).

that are essential for function during conformational plasticity. For example, selected with a red rectangle are contacts which appear in the contact map only for the closed, ligand bound, conformation (Fig. 3.4). By integrating structural information taken from different states, DI pairs can be used to identify an ensemble of different conformation of the protein along with their coevolutionary signals. The multiple state model of DCA-MOL will allow users to clearly visualise the interactions during protein dynamics.

3.3.3. PConsFam – a database of DCA-based structure predictions

Since its inception DCA was used to predict various structural properties of proteins, however due to the statistical machinery used, the results were only significant for a

limited number of protein families, with sufficiently large number of diverse members. By the addition of deep learning methodologies it is possible to improve the DI based contact predictions, to get reasonable results even for proteins with only a few hundred members. Results of such prediction for more than 13000 protein families from the PFam database (El-Gebali *et al.*, 2018) can be found in PConsFam (Lamb *et al.*, 2019). More than a half of those proteins have no known reference structure yet.

When modeling protein structures it is important to be able to recognise strengths and weaknesses of the model. In particular in case of *de novo* predictions, for proteins with no known homologues, it is important to consider what assumptions where used to create it. In case of DCA-based prediction, these assumptions are the pairs selected as relevant (co-evolving) based on their Direct Information scores.

User interface in the PConsFam database (Fig. 3.5) guides the user through the analysis of the structure, by allowing multiple modes of comparison between the DI scores for the whole alignment of a protein family and the resulting structure. It is important to note, that to ensure high quality of the results parts of the multiple sequence alignment may be discarded (e.g. if there are too many gaps). When a reference (e.g. crystal) structure for the protein is available, the parts corresponding to the modeled region are presented to the user for comparison.

To ensure that a proper mapping between the model and the reference structure is created, a global alignment is calculated (using a modified Needleman-Wunsch algorithm – Alg. 5) – as the sequences should in principle be identical, but both may contain unresolved regions, the information of backbone breaks is used for proper gap placement in the alignment (alignment is done by aligning each structure to the known full sequence, and extracting corresponding columns). More details about the applications of PConsFam can be found in (Lamb *et al.*, 2019).

3.3.4. Prediction of minimal interactions for protein folding

Knotted proteins remain a mystery in many aspects. Chief among them is the one connected to the long held belief (Mansfield, 1994), that protein chains cannot be knotted – how do they fold? Protein folding is in a large part dependant on the hydrophobic collapse, and excluded volume effect, both of which should increase knotting rates in long polymers – except for the multitude of inter-residue interactions within a protein structure. Smaller loop clearance in a tightly packed structure makes it much less probable for a chain to cross the loop surface without any additional driving force.

This crossing is assumed to be a rate limiting step in knotted protein folding (King, 2010), and one of the reasons why computational studies of the folding process are problematic. Knotted proteins are for the most part too large for all-atom explicit

solvent molecular dynamics (MD) simulations, and simplified, structure-based coarse-grained models (Levitt and Warshel, 1975; Kmiecik *et al.*, 2016) do not fold the knotted proteins reliably enough to fully explore their free energy landscape. Coarse grained models are structure approximations where each residue is represented by one (e.g. in G$\bar{o}$-like models (Taketomi *et al.*, 1975)) or more beads (UNRES force field (Sieradzan *et al.*, 2015), CABS force field (Kmiecik *et al.*, 2011)). In structure based models, the potential energy of any given conformation is calculated in relation to the native (reference) structure of the protein. Potential energy of the structure, which is minimised during the folding simulation is given by a Hamiltonian (Clementi *et al.*, 2000):

$$\begin{aligned} E(\Gamma,\Gamma_0) = &\sum_{bonds} K_r(r-r_0)^2 + \sum_{angles} K_\theta(\theta-\theta_0)^2 \\ &+ \sum_{dihedral} K_\phi^{(n)}[1+cos(n(\phi-\phi_0))] \\ &+ \sum_{i<j-3}\left\{\varepsilon_{ij}[5(\frac{\sigma_{ij}}{r_{ij}})^{12}-6(\frac{\sigma_{ij}}{r_{ij}})^{10}]+\varepsilon'_{ij}(\frac{\sigma_{ij}}{r_{ij}})^{12}\right\}, \end{aligned} \tag{3.7}$$

where Γ is the current coordinate configuration and Γ_0 is the native (reference) structure. Respectively, r and r_0 represent the distance between beads representing consecutive residues, θ and θ_0 the angle formed by three consecutive beads, and ϕ and ϕ_0 the dihedral angle between four consecutive beads, in the current configuration and the reference structure. The dihedral potential consists of a sum of two terms for every four adjacent C-α atoms, one with period n = 1 and one with n = 3. Those three terms represent the geometry of the backbone (and can in fact be modified to fold a protein (Najafi and Potestio, 2015)), however, the most interesting force behind the folding is the last term. It involves non-local native interactions (hence the $i < j-3$ in the sum) and is calculated as a Lenard-Jones (Jones, 1924) potential with attractive and repulsive terms with the minimum at the distance found in the native structure (σ variables), where ε is the depth of the potential well. K parameters are weights to modify the relative strength of interactions and depend on the force field used. There are some limitations to the Lenard-Jones potential, in particular the excluded volume effect. Implicit size of the bead in a pair depends on the native distance, and this can have a large effect on the entropy of a folded state. In our work we used instead an attractive Gaussian well

$$C_G(r_{ij}, r_0^{ij}) = (1+(\frac{\sigma_{NC}}{r_ij}{}^1 2)(1-\exp(\frac{-(r_{ih}-r_0^{ij})^2}{2\sigma^2}))-1,$$

where σ is derived based on the Lenard-Jones potential (Noel and Onuchic, 2012), and

$$\sigma^2 = \frac{(r_0^{ij})^2}{50\ln 2}.$$

Modification to the last term in the equation, by e.g. adding attraction between non-native residue pairs, allows for inclusion of additional driving forces to the simulation (Wallin *et al.*, 2007).

Analysis described in (Dabrowski-Tumanski *et al.*, 2015) is an attempt to identify the forces that help knotted proteins to in folding, by combining MD simulations with other bioinformatics tools to identify the minimal set of physical interactions necessary for entangling a protein (in this case the smallest knotted proteins with PDB Id 2efv). As the "conventional" means – native structure-based contacts and chemical and physical properties-based non-native interactions (Miller *et al.*, 1987; Miyazawa and Jernigan, 1996; Sippl, 1990) – of studying the folding process of large molecules fail against the conundrum of knotting, a new criterion is introduced, based on sequence co-evolution.

Our main idea was to identify regions of the structure that appear to interact, exhibiting as a joint evolutionary pressure, but be far enough in the final structure, that it will not explain such co-dependency. A natural assumption here, is that those contact are important in transition states, and facilitate the correct folding pathway.

Two clusters of DCA-determined contacts with highest DI score were selected, with the assumption, that in folding two interacting neighbourhoods of positions are more probable to have a noticeable influence, than a single point of contact. We then added them as non-native interactions to the coarse-grained folding simulation, and both appeared to smooth the folding landscape, which suggests that knotting is one of factors imposing evolutionary pressure on the surrounding structure.

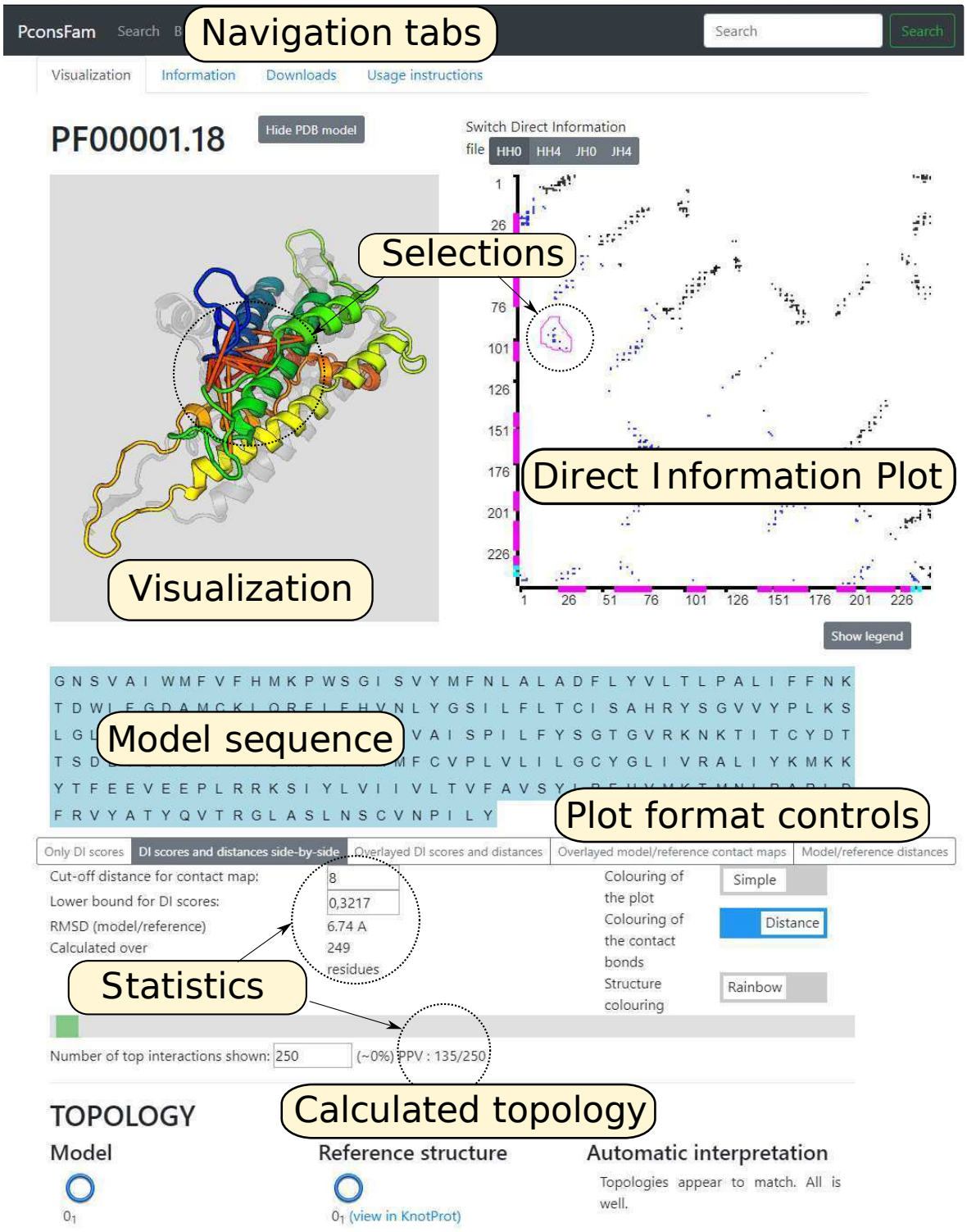

Figure 3.5: User interface of PConsFam detailing results for the PF00001 family. The default Visualization's tab contains structure visualisation of the model(s) (superposed with reference structure if available), Direct Information (DI) plot (which can also display contact maps), and the sequence and topology of the models. Range and format of displayed contacts can be changed, and contacts between residues can be visualised as bonds on the structure. RMSD between model and reference, and a positive predicted value score (PPV) indicating overlap between residues pairs and structural contacts in the model are also shown. Other tabs contain additional information about the family, and download links for calculated data.

"There once was a protein knot,
that folded itself without thought.
Chaperonins paled,
simulations failed,
and researchers lost all the plot!"

Nodus Anonymus

4

Databases and algorithmic tools for protein topology explorations

TOPOLOGY, by its very name (from the Greek $\tau o \pi o \zeta$(*topos*), place, and $\lambda o \gamma o \zeta$(*logos*), study), is a branch of mathematics studying the properties of space, and it's deformations. In the case of proteins, "topology" has been used to describe relative placement of secondary structure elements in the structure of a molecule (Rawlings *et al.*, 1985), however since the discovery of knot-like entanglements along the protein chains in 1994 (Mansfield, 1994) it has been getting more traction in referring to folds similar to those described mathematically.

This chapter presents works published as (Dabrowski-Tumanski *et al.*, 2016a) (Section 4.1.1), (Jarmolinska *et al.*, 2018) (Section 4.2), and (Jarmolinska *et al.*, 2019a) (Section 4.3).

4.1. Databases collecting information about topologically complex structures

While the existence of knot-like structures in proteins has been shown a quarter of a century ago, only in the recent years has the number of known protein structures, including those entangled risen to the levels which prompted the creation of servers

and databases dedicated to the study of topological complexities of biomolecules.

4.1.1. LinkProt: a database collecting information about biological links

Entanglements of a single protein chain have been studied for more than 20 years, but only recently the acceptance of those chains as knots has been taken to its natural conclusion. If each chain is a, possibly trivial, knot, then a protein complex forms an, again possibly trivial, link. Some actually connected multichain arrangments, have been, pardon the pun, linked to the function of proteins in question. Examples include domain-swapped proteins (Baiesi *et al.*, 2016), and catenates forming viral capsids (Duda, 1998). An important consideration here is that protein chains are open curves. Thus, similar to the case of knot-like structures, first an implicit closing curve must be added to connect the termini. Then the chain backbone is a proper knot, and can be analysed to find possible links. However, protein links can also be defined within a single chain (Liang and Mislow, 1995; Boutz *et al.*, 2007). Protein backbone forms loops, which can be occasionally closed through a covalent bond – such as a cysteine bridge (interaction between sulphur atoms found in some of the amino acid residues). Such loop can be considered a closed curve – a valid mathematical knot – and in proteins where more than one such closed loop exists – a part of a valid link (Fig. 4.1). To allow researchers an easy and intuitive access to the information about various types of link-like structures found in proteins we have a created a self-updating database of biological links – LinkProt (Dabrowski-Tumanski *et al.*, 2016a) (available at `https://linkprot.cent.uw.edu.pl`).

All entries in the database are based on structures available in RCSB PDB database, and our database is updated each week, as the new structures in PDB are released. For the link type identification the components are represented by the coordinates of the $C-\alpha$ atoms (with the gaps in the structure filled with straight line segments between present atoms). In both cases all the combinations of deterministic or probabilistic components (up to 4 at a time) are given to the link identification algorithm, with an addendum for whole chain based components. For two chains to be considered as a potential non trivial link, there must be at least one coordinate of one of the structures which is closer to the centre of mass of the second structure than any of its own atoms – as link type determination in a non-deterministic case is the most computationally heavy step, this modification significantly speeds up the calculations in multichain structures. To find out how exactly are components entangled a *minimal surface* is computed for each component (Niemyska *et al.*, 2016; Dabrowski-Tumanski *et al.*, 2016b). Link type is identified using the Ewing-Millet (Ewing and Millett, 1991)

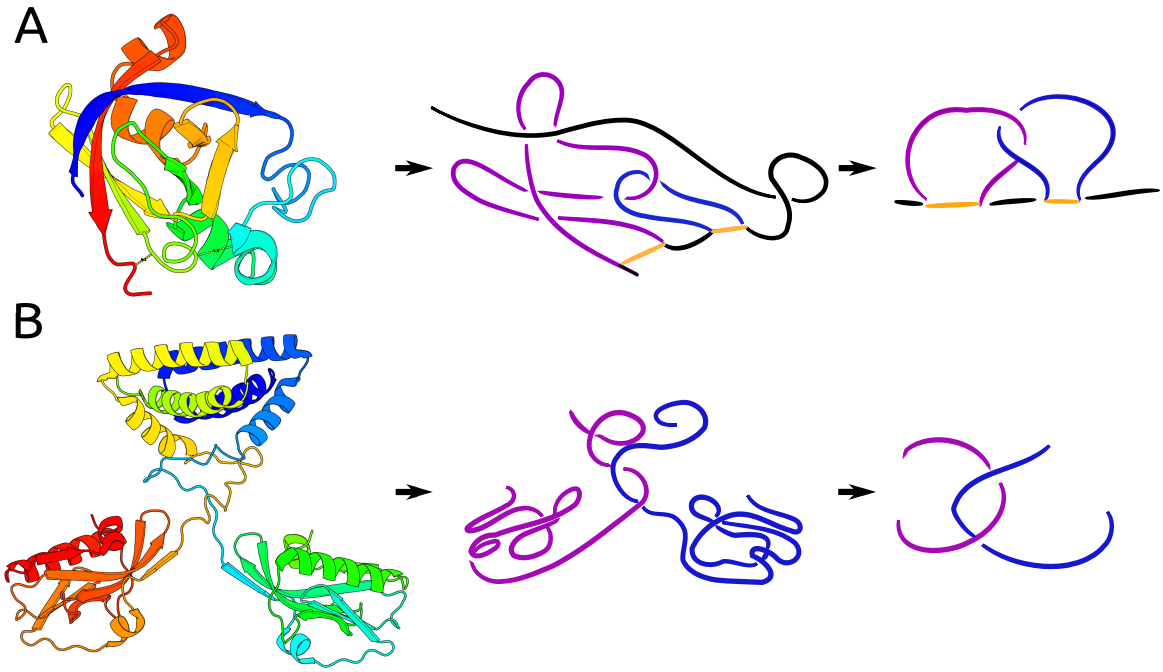

Figure 4.1: Examples of Hopf links found in proteins: structure visualisation (using PyMOL), topology sketch, simplified topology sketch: (A) deterministic (PDB Id 1bw3 chain A); (B) probabilistic (PDB Id 5nt2 chains D and E).

implementation of the HOMFLY-PT polynomial as it distinguishes not only classes, but also chirality and orientation of the link.

Additionally, each newly added link is compared to those already present in database to update the non-redundancy table. For single protein chain non-redundancy is often defined as cutoff of sequence identity of e.g. 30%. In case of links, which may consist of multiple chains the comparison is made by finding for each chain from one link its sequentially closest counterpart in the second link, without repetitions (two links with different number of chains are non-redundant by default). Then for each pair of chains the 30% identity criterion is verified.

Underlying database was designed to minimise the amount of data necessary, while keeping the correct separation to allow a single protein chain to contain deterministic links and potentially be part of multiple linked sets of chains. Database scheme is presented in Fig. 4.2.

User interface

To facilitate the analysis of data available multiple filters can be applied to the non-trivially linked protein data set. Proteins can be grouped by the topology, but also family to which they belong, enzyme classification, keywords from the PDB description, and filtered so that only a non redundant sequences are shown. Redundancy here signifies the sequence identity cutoff above which a protein is considered to be

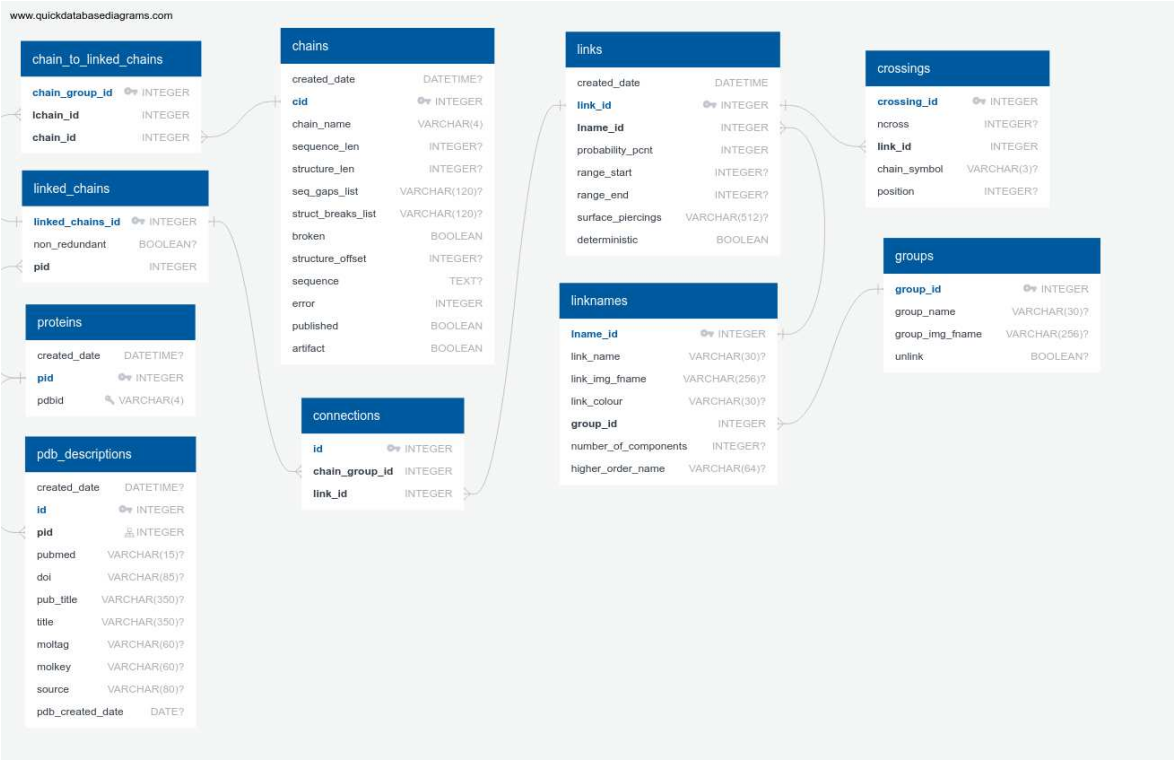

Figure 4.2: Key tables of the database schema designed for LinkProt.

sufficiently represented by another. As the actual entities of the database are links, the redundancy takes into account similarity between most similar pairs of component sequences between two links.

Each single entry page presents the results of one link – that is one combination of components present in a given, possibly multichain, protein. In case of deterministic links (that is those with curves closed by covalent bonds) this is a straight forward result, with always only one link type assigned. In case of non-deterministic (here called "probabilistic") links, due to the random nature of component closing attempts a number of possible link types can be obtained for the same components. Then all of them are presented, with their probability of belonging to a given class of link types specified. All links for which the probability of an unlink topology was less than 60% are included in the database.

4.2. GapRepairer – a server for topologically conscious reconstruction of missing parts of protein models

Over a quarter of known protein structures possess unresolved fragments – which hinders the study of afflicted proteins in many ways, in particular in regards to their

topology. When determining the entanglements found on a protein chain, gaps are usually filled in as a straight line connecting the last residues on both sides, which often results in incorrect topology assignment. However, topology is not usually verified by software designed for protein structure modeling. GapRepairer (Jarmolinska *et al.*, 2018) is a tool we created to bridge this gap, and allow easy repair and modification of protein structures.

GapRepairer was designed as a user-friendly homology modeling tool, although thanks to the capabilities of MODELLER (Webb and Sali, 2014) software, it will also attempt to fill any gaps without reference structure *de novo*. To monitor the topology of the protein, both potential templates and resulting models are screened for inconsistencies in knot type – as protein topology has been shown to be better conserved than the sequence (Sułkowska *et al.*, 2012b).

Applications of GapRepairer

The entanglement appears in protein modeling in various ways giving rise to many applications of GapRepairer. The most fundamental is topologically valid gap modeling. The server may also be used to assess the topological correctness of the structure. This is especially important in the case of low-resolution structures (such as coming from CryoEM) and in structure prediction competitions (e.g. CASP (Moult *et al.*, 1995), CAPRI (Janin *et al.*, 2003)). Another application is to generate a topologically valid list of homologues (either structural or sequential). Finally, the user can utilise non-entanglement-based functions of GapRepairer unique to this server, such as the modeling of a chain in the neighbourhood of other chains (important for domain-swapped structures), or modeling chains with exceptionally large gaps.

Possible uses for GapRepairer include:

- **Topologically valid gap modeling.** For some structures, the easiest way of modeling is not the best one, as it can artificially (dis)entangle the target. E.g. the original structure of glycohydrolase (PDB Id 3sij) possesses a deep $+3_1$ knot. However, the non-trivial topology stems from the location of the 9-residue-long gap. Modeling this gap with GapRepairer correctly disentangles the model (Fig. 4.3A). Similarly, the hydrolase with PDB Id 2d7d, after a proper modeling changes topology from deep 4_1 knot to unknot. Modeling CryoEM structure of human 26S proteasome (resolution 6.8 Å PDB Id 5ln3, chain N) also disentangles the protein. The modeling can also change the chirality of the knot and reduce the complex topological fingerprint, as in the case of human hydrolase with PDB Id 4zg6. The gapped structure features a left-handed -3_1 knot, but after gap filling, it turns out that it has the right-handed $+3_1$ knot (Fig. 4.3B).

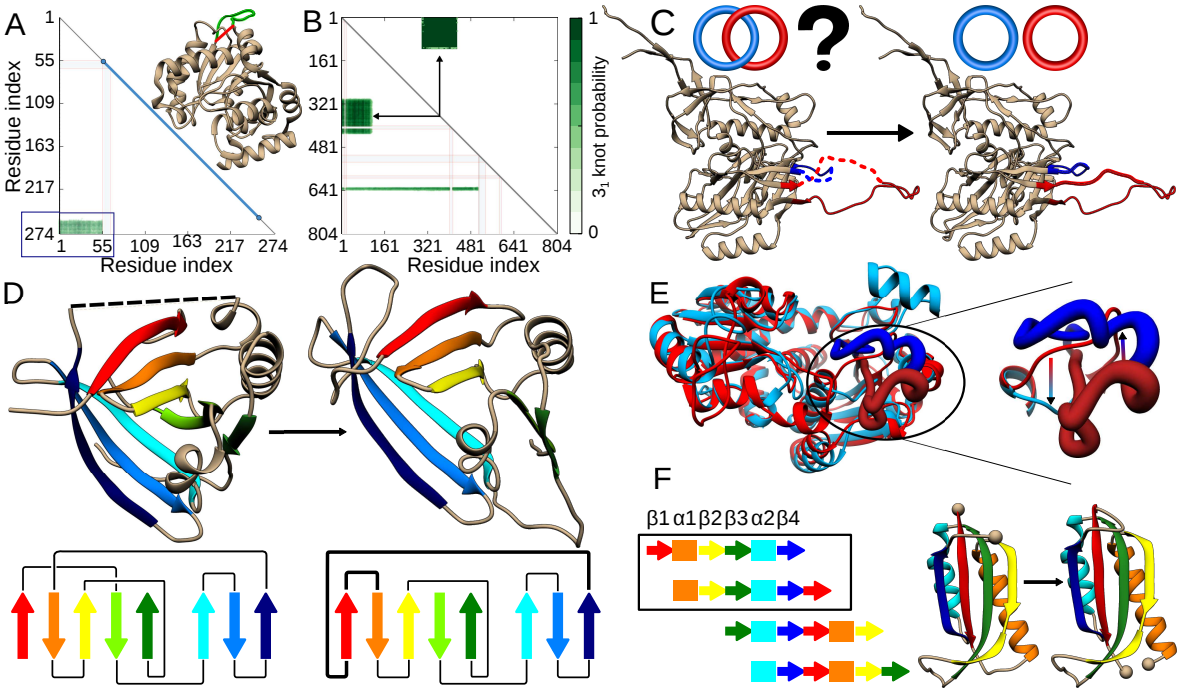

Figure 4.3: Possible utilization of GapRepairer server. (A) Disentangling of an artificially knotted protein with PDB Id 3SIJ—the straight interval joining gap ends (red stripe in the structure) results in a 3_1 knotted protein, as shown in the matrix fingerprint (highlighted by blue rectangle). The correct way of gap modeling is shown with the green curve on the structure. (B) Change in the fingerprint and topology upon gap modeling for the protein with PDB Id 4zg6. Before modeling (below diagonal) two knotted regions can be spotted. After gap modeling (above diagonal), only one portion of the chain is 3_1 knotted. (C) Assessing topological correctness. For the potentially Hopf-linked protein with PDB Id 3j70, removing and remodeling of parts of the loops (dashed lines in left panel), results in unlinked loops (right panel). The topology in each case is shown schematically above the structure. (D) Validating crystallographic data—remodeling parts of the protein with PDB Id 2xkl (left panel) reveals an incorrect connection between b-strands (right panel). For each case, the scheme of b-strands connection is shown below the structure. (E) Search for a topologically valid template—the structures of ATC (red) and OTC (blue) are almost perfect structural homologues, yet they differ in the location of pieces of chain in one part, shown as the thick structure and enlarged in the right panel. Interchanging the parts according to the arrows shown in the right panel changes the topology of the protein. (F) The idea of circular permutation of protein fragments. The right panel shows exemplary structures corresponding to the scheme in the frame.

Even though GapRepairer is not optimized to reconstruct membrane proteins, it properly models loops based on structural and topological assumptions when homological chains are unknown. E.g. modeling amino acid transporter (PDB Id 5llm, chain A, 4 gaps up to 23 amino acids) changes its topology from knot to slipknot characteristic to all of the members of Sodium-dicarboxylate symporter family. Another membrane protein (PDB Id 5l25) has 6 gaps in total, one comprising 94 residues. In this case, careless modeling generates the complex topological fingerprint, with unnatural 8_2 knot. Proper modeling of this loop results in regular $S3_13_1$ slipknot motif. Furthermore, reconstruction (using only structural homologues) of the membrane protein with PDB Id 4kjs chain A reveals the knot and therefore the first deeply knotted membrane protein family (Jarmolinska *et al.*, 2019a).

- **Assessing the topological correctness** Assessing the topological correctness is crucial for already known structures with dubious fragments, e.g. coming from low resolution techniques, such as CryoEM. For example, the human immune system related protein (PDB Id 3J70, from CryoEM) contains, according to the model, two linked covalent loops. However, after cutting out a part of each loop and remodeling, the loops turn out to be unlinked, which stays in accordance with all experimentally determined homologues (Fig. 4.3C). Similarly, the structure of methyltransferase (PDB Id 1oy5) is unknotted, although all proteins from this family form a deeply $+3_1$ knotted structure. After the remodeling of this protein, a structure with a deep $+3_1$ knot is recreated. Finally, the covalent loop placement may also disclose improperly crystallized protein. Remodeling of the murine lipid transport protein (PDB Id 2xkl) reveals that the -sheets in the crystal structure were joined in a wrong order (Fig. 4.3D). Such topological assessment is decisive in prescribing the function and the properties of a protein, especially as the function of a protein can be guessed knowing the function characteristic for such topological motif (Niemyska *et al.*, 2016; Sułkowska *et al.*, 2012b).

- **Redesigning protein architecture**. The GapRepairer can be also used to design a new architecture of the proteins, e.g. to design entangled proteins, or disentangle the topologically non-trivial ones by the addition of a loose loop. This can be a useful technique in understanding the role of knots (King, 2010; Yeates *et al.*, 2007) (by comparison of different topology homologues). The ability to design topologically non-trivial structures can be as well used to engineer highly stabilized proteins (Ghosh *et al.*, 2015) or to design intradomain linkers (Scalley-Kim *et al.*, 2003). Template selection is fundamental in any homology modeling or homology-based function prediction. However, even the close homologues can differ in topology as e.g. the pair Acetylornithine TransCarbamylase (knotted) and Ornithine TransCarbamylase (ATC/OTC) (unknotted). Only GapRepairer, using its topological analysis can distinguish these cases and select relevant close homologues (Fig. 4.3E).

- **Other uses.** GapRepairer is also equipped with functions not necessarily connected with the protein topology. The ability to repair one chain in the presence of the second makes it a unique on-line tool capable of treating domain-swapped proteins. It also allows GapRepairer to model gaps in many-chain systems. The GapRepairer has no restriction on the gap size if only enough homologues are present. This allows one to model a structure with circularly permuted fragments of the protein (Fig. 4.3F), one of the techniques used to investigate the protein energy landscape (Lindberg, 2006). Finally, because of the DALI database (Holm

and Laakso, 2016) utilization, GapRepairer is the only webserver which can cope with structures with no sequential homologues, if there are some structurally similar chains.

An in depth exploration of those use cases can be found in the online documentation (`https://gaprepairer.cent.uw.edu.pl`).

4.3. Diversity of knotted proteins

Folding, which is the process in which a newly translated protein reaches its final shape, is a non-trivial task. Only by attaining its proper structure can a molecule function as it should, which makes folding an essential process – hence a complex cellular machinery present to oversee it. Even topologically trivial proteins often need help in folding, provided e.g. by other proteins called molecular chaperons. However, in case of knotted proteins the process is even more complicated, as for a knot to appear the chain of a folding protein must make a, at least once, twisted loop, stable enough for another loop, or protein terminus, to thread through. The exact mechanisms which help this process in a crowded, cellular environment are unknown – proposed answers include binding to the translating ribosome and forces provided by non-native interactions in the chain (see Section 3.3.4). Nonetheless, it is possible to give some intuition about the possible folding pathways of knotted proteins using the traditional approach to *in silico* studies of molecular folding, that is through coarse grained molecular dynamics simulations. In this approach each amino acid is replaced by a single bead, which interactions with all the other bead are defined based on the proximity in the known final structure.

In (Jarmolinska *et al.*, 2019a) we propose the folding pathways of 3 knotted proteins structures:

1. a mitochondiral apoptosis-inducing factor (PDB Id 5fmh, chain a),
2. a protein from the large subunit of the human mitochondrial ribosome (PDB Id 4v1a, chain w), and
3. a protein of unknown function from *Treponema pallidum* (PDB Id 5jir, chain A).

Based on the simulations, mitochondiral apoptosis-inducing factor (Aifm1) can be divided into four stuctural domains, folding independently. The knot is fully contained in the N-terminal domain, which allows it to fold at any point during the whole folding process, and can in fact be facilitated by one of the domain termini being held in place by the already folded rest of the structure.

Both of the other proteins studied need the proper terminus threading to be the first step in folding, otherwise there is a high chance of a topological trap prohibiting acquiring the proper structure. The ribosomal protein has a complex topological fingerprint involving multiple slipknots – the core of the structure is made up of several loops wrapped around by another, twisted, loop – held together by a shallow threading of the C terminus which results in a knot. Simulations show that wrapping the outer loop around the protein poses no problem, however the only way to fold correctly is for the N terminus to first thread through this loop, followed by the C terminus (which makes the structure knotted) and the internal loop (which results in the slipknots). Any other order of threading prohibits the N terminus from plugging through the outer loop and locks the protein in a topological trap.

The most interesting case studied is the deeply embedded knot in the *T. pallidum* protein. It is made up of 3 separate domains, each over 140 residues long, with a 3_1 knot in the middle domain. The only way for this protein to fold is if the twisted loop forms at the very beginning and one of the termini threads through it soon after. Otherwise, folding of the two outer domains exerts constant tension on the middle part of the protein which prohibits folding. Additionally, it two terminal domains were folded first, it would be impossible for the middle domain to provide a loop large enough for one of those domains to pass through.

"The road goes ever on and on"

J.R.R. Tolkien, The Hobbit

"Wszyscy zgodnie orzekli, że nigdy nie będą brali ze sobą żadnych map, bo najciekawsze są podróże w nieznane."

Narrator, odcinek 5 serialu Muminki

5 Summary

This dissertation presents several different approaches to the study of proteins, focusing mostly on those relating to the topology of their structures, with a small detour into the co-evolutionary methods, as they are still relatively unexplored, and thus promising, as an avenue for discovering secrets of biomolecular entanglements. First, a new, and the first deterministic, algorithm knot_pull for entanglement analysis of biological molecules is introduced. Then, a novel algorithm for multiple profile alignment is presented, and applied to provide the first study on the evolution of slipknotted membrane proteins. Different resources for exploration (DCA-MOL) and exploring (PConsFam) the structural information provided by sequence positions co-evolution are described, and the interactions found used for knotted protein folding simulations. Then, a self-updating database of links in proteins LinkProt is shown, along with a webserver for topologically-conscious modeling of missing parts of protein structure coordinates. Finally, possible folding pathways of newly discovered entangled proteins are explored.

Compared to many other fields of protein research, the study of topology is barely crawling – there are still many more questions than answers, including the fundamental ones:

> Why did the knots emerge during evolution? Was it purely a random mutation that stuck, or did they bring some crucial advantage?

How did they appear? Was each knot that remains in genome a one time event, or did those knots deepen, or increase their complexity, over time?

What makes them fold efficiently, or at least efficiently enough?

Is there a method to the complexity of knots found in proteins? What is the reason for different knot types and chiralities found in protein structures?

Is there even a common to all knots answer to this questions?

Works presented here attempt to bring us closer to the answers but there is still a lot more to do, hence the main goal of most of the research presented was to allow others a smoother pass into the mysterious world of biological entanglements.

Bibliography

Alexander, J. W. (1928). Topological invariants of knots and links. *Transactions of the American Mathematical Society*, **30** (2), 275–306.

Altschuh, D., Lesk, A., Bloomer, A. and Klug, A. (1987). Correlation of co-ordinated amino acid substitutions with function in viruses related to tobacco mosaic virus. *Journal of Molecular Biology*, **193** (4), 693–707.

Arlazarov, V. L., Dinitz, Y. A., Kronrod, M. and Faradzhev, I. (1970). On economical construction of the transitive closure of an oriented graph. In *Doklady Akademii Nauk*, Russian Academy of Sciences, vol. 194, pp. 487–488.

Baiesi, M., Orlandini, E., Trovato, A. and Seno, F. (2016). Linking in domain-swapped protein dimers. *Scientific Reports*, **6**, 33872.

Baldassi, C., Zamparo, M., Feinauer, C., Procaccini, A., Zecchina, R., Weigt, M. and Pagnani, A. (2014). Fast and accurate multivariate gaussian modeling of protein families: predicting residue contacts and protein-interaction partners. *PloS One*, **9** (3), e92721.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. and Bourne, P. E. (2000). The protein data bank. *Nucleic Acids Research*, **28** (1), 235–242.

Bitbol, A.-F., Dwyer, R. S., Colwell, L. J. and Wingreen, N. S. (2016). Inferring interaction partners from protein sequences. *Proceedings of the National Academy of Sciences*, **113** (43), 12180–12185.

Bölinger, D., Sułkowska, J. I., Hsu, H.-P., Mirny, L. A., Kardar, M., Onuchic, J. N. and Virnau, P. (2010). A stevedore's protein knot. *PLoS Computational Biology*, **6** (4), e1000731.

Boutz, D. R., Cascio, D., Whitelegge, J., Perry, L. J. and Yeates, T. O. (2007). Discovery of a thermophilic protein complex stabilized by topologically interlinked chains. *Journal of Molecular Biology*, **368** (5), 1332–1344.

Brudno, M., Malde, S., Poliakov, A., Do, C. B., Couronne, O., Dubchak, I. and Batzoglou, S. (2003). Glocal alignment: finding rearrangements during alignment. *Bioinformatics*, **19** (suppl_1), i54–i62.

Bryant, T., Watson, H. and Wendell, P. (1974). Structure of yeast phosphoglycerate kinase. *Nature*, **247** (5435), 14.

Callaway, D. J. (1994). Solvent-induced organization: A physical model of folding myoglobin. *Proteins: Structure, Function, and Bioinformatics*, **20** (2), 124–138.

Chan, H. S. and Dill, K. A. (1993). The protein folding problem. *Physics Today*, **46** (2), 24–32.

Cheng, R. R., Morcos, F., Levine, H. and Onuchic, J. N. (2014). Toward rationally redesigning bacterial two-component signaling systems using coevolutionary information. *Proceedings of the National Academy of Sciences*, **111** (5), E563–E571.

Claverie, J.-M. and Notredame, C. (2006). *Bioinformatics for dummies*. John Wiley & Sons.

Clementi, C., Nymeyer, H. and Onuchic, J. N. (2000). Topological and energetic factors: what determines the structural details of the transition state ensemble and "en-route" intermediates for protein folding? an investigation for small globular proteins. *Journal of Molecular Biology*, **298** (5), 937–953.

Consortium, U. (2018). Uniprot: a worldwide hub of protein knowledge. *Nucleic Acids Research*, **47** (D1), D506–D515.

Crooks, G. E., Hon, G., Chandonia, J.-M. and Brenner, S. E. (2004). Weblogo: a sequence logo generator. *Genome Research*, **14** (6), 1188–1190.

Dabrowski-Tumanski, P., Jarmolinska, A. and Sulkowska, J. (2015). Prediction of the optimal set of contacts to fold the smallest knotted protein. *Journal of Physics: Condensed Matter*, **27** (35), 354109.

—, Jarmolinska, A. I., Niemyska, W., Rawdon, E. J., Millett, K. C. and Sulkowska, J. I. (2016a). Linkprot: A database collecting information about biological links. *Nucleic Acids Research*, **45** (D1), D243–D249.

—, Niemyska, W., Pasznik, P. and Sulkowska, J. I. (2016b). Lassoprot: server to analyze biopolymers with lassos. *Nucleic Acids Research*, **44** (W1), W383–W389.

Dayhoff, M., Schwartz, R. and Orcutt, B. (1978). A model of evolutionary change in proteins. In *Atlas of Protein Sequence and Structure*, vol. 5, National Biomedical Research Foundation Silver Spring, pp. 345–352.

De Leonardis, E., Lutz, B., Ratz, S., Cocco, S., Monasson, R., Schug, A. and Weigt, M. (2015). Direct-coupling analysis of nucleotide coevolution facilitates rna secondary and tertiary structure prediction. *Nucleic Acids Research*, **43** (21), 10444–10455.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, **1** (1), 269–271.

Dos Santos, R. N., Morcos, F., Jana, B., Andricopulo, A. D. and Onuchic, J. N. (2015). Dimeric interactions and complex formation using direct coevolutionary couplings. *Scientific Reports*, **5**, 13652.

Dowker, C. H. and Thistlethwaite, M. B. (1983). Classification of knot projections. *Topology and its Applications*, **16** (1), 19–31.

Duda, R. L. (1998). Protein chainmail: catenated protein in viral capsids. *Cell*, **94** (1), 55–60.

Duran, A. M. and Meiler, J. (2013). Inverted topologies in membrane proteins: a mini-review. *Computational and Structural Biotechnology Journal*, **8** (11), e201308004.

Eddy, S. R. (1998). Profile hidden markov models. *Bioinformatics*, **14** (9), 755–763.

Ekeberg, M., Lövkvist, C., Lan, Y., Weigt, M. and Aurell, E. (2013). Improved contact prediction in proteins: using pseudolikelihoods to infer potts models. *Physical Review E*, **87** (1), 012707.

El-Gebali, S., Mistry, J., Bateman, A., Eddy, S. R., Luciani, A., Potter, S. C., Qureshi, M., Richardson, L. J., Salazar, G. A., Smart, A. *et al.* (2018). The pfam protein families database in 2019. *Nucleic Acids Research*, **47** (D1), D427–D432.

Ewing, B. and Millett, K. C. (1991). A load balanced algorithm for the calculation of the polynomial knot and link invariants. In *The Mathematical Heritage of CF Gauss*, World Scientific, pp. 225–266.

Finn, R. D. e. a. (2015). The pfam protein families database: towards a more sustainable future. *Nucleic Acids Research*, **44** (D1), D279–D285.

Freyd, P., Yetter, D., Hoste, J., Lickorish, W. R., Millett, K. and Ocneanu, A. (1990). A new polynomial invariant of knots and links. In *New Developments In The Theory Of Knots*, World Scientific, pp. 12–19.

Ghosh, E., Kumari, P., Jaiman, D. and Shukla, A. K. (2015). Methodological advances: the unsung heroes of the gpcr structural revolution. *Nature Reviews Molecular Cell Biology*, **16** (2), 69–81.

Gibbs, A. J. and McIntyre, G. A. (1970). The diagram, a method for comparing sequences: Its use with amino acid and nucleotide sequences. *European Journal of Biochemistry*, **16** (1), 1–11.

Göbel, U., Sander, C., Schneider, R. and Valencia, A. (1994). Correlated mutations and residue contacts in proteins. *Proteins: Structure, Function, and Bioinformatics*, **18** (4), 309–317.

Hass, J., Lagarias, J. C. and Pippenger, N. (1999). The computational complexity of knot and link problems. *Journal of ACM*, **46** (2), 185–211.

Heidrich, W. (2005). Computing the barycentric coordinates of a projected point. *Journal of Graphics Tools*, **10** (3), 9–12.

Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, **89** (22), 10915–10919.

Henrich, A. and Kauffman, L. H. (2014). Unknotting unknots. *The American Mathematical Monthly*, **121** (5), 379–390.

Hirschberg, D. S. (1975). A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, **18** (6), 341–343.

Holm, L. and Laakso, L. M. (2016). Dali server update. *Nucleic Acids Research*, **44** (W1), W351–W355.

Huelsenbeck, J. P. and Ronquist, F. (2001). Mrbayes: Bayesian inference of phylogenetic trees. *Bioinformatics*, **17** (8), 754–755.

Jamroz, M., Niemyska, W., Rawdon, E. J., Stasiak, A., Millett, K. C., Sułkowski, P. and Sulkowska, J. I. (2014). Knotprot: a database of proteins with knots and slipknots. *Nucleic Acids Research*, **43** (D1), D306–D314.

Janin, J., Henrick, K., Moult, J., Eyck, L. T., Sternberg, M. J., Vajda, S., Vakser, I. and Wodak, S. J. (2003). Capri: a critical assessment of predicted interactions. *Proteins: Structure, Function, and Bioinformatics*, **52** (1), 2–9.

Jarmolinska, A. I., Kadlof, M., Dabrowski-Tumanski, P. and Sulkowska, J. I. (2018). Gaprepairer: a server to model a structural gap and validate it using topological analysis. *Bioinformatics*, **34** (19), 3300–3307.

—, Perlinska, A. P., Runkel, R., Trefz, B., Ginn, H. M., Virnau, P. and Sulkowska, J. I. (2019a). Proteins' knotty problems. *Journal of Molecular Biology*, **431** (2), 244–257.

—, Zhou, Q., Sulkowska, J. I. and Morcos, F. (2019b). Dca-mol: A pymol plugin to analyze direct evolutionary couplings. *Journal of Chemical Information and Modeling*, **59** (2), 625–629.

Jones, A. (1985). A polynomial invariant for knots via von neumann algebras. *Bulletin of the American Mathematical Society*, **12** (1), 103.

Jones, J. E. (1924). On the determination of molecular fields.—ii. from the equation of state of a gas. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, **106** (738), 463–477.

Kececioglu, J. (1993). The maximum weight trace problem in multiple sequence alignment. In *Combinatorial Pattern Matching*, Springer Berlin Heidelberg, pp. 106–119.

King, N. P. t. a. (2010). Structure and folding of a designed knotted protein. *Proceedings of the National Academy of Sciences*, **107** (48), 20732–20737.

Kmiecik, S., Gront, D., Kolinski, M., Wieteska, L., Dawid, A. E. and Kolinski, A. (2016). Coarse-grained protein models and their applications. *Chemical Reviews*, **116** (14), 7898–7936.

—, Jamroz, M. and Kolinski, A. (2011). Multiscale approach to protein folding dynamics. In *Multiscale Approaches to Protein Modeling*, Springer, pp. 281–293.

Koniaris, K. and Muthukumar, M. (1991). Self-entanglement in ring polymers. *The Journal of Chemical Physics*, **95** (4), 2873–2881.

Lackenby, M. (2015). A polynomial upper bound on reidemeister moves. *Annals of Mathematics*, **182** (2), 491–564.

— (2016). The efficient certification of knottedness and thurston norm. *arXiv preprint arXiv:1604.00290*.

Lamb, J., Jarmolinska, A. I., Michel, M., Menéndez-Hurtado, D., Sulkowska, J. I. and Elofsson, A. (2019). Pconsfam: An interactive database of structure predictions of pfam families. *Journal of Molecular Biology*, **431** (13), 2442–2448.

Levitt, M. (1976). A simplified representation of protein conformations for rapid simulation of protein folding. *Journal of Molecular Biology*, **104** (1), 59–107.

— and Warshel, A. (1975). Computer simulation of protein folding. *Nature*, **253** (5494), 694.

Liang, C. and Mislow, K. (1995). Topological features of protein structures: knots and links. *Journal of the American Chemical Society*, **117** (15), 4201–4213.

Lim, K., Zhang, H., Tempczyk, A., Krajewski, W., Bonander, N., Toedt, J., Howard, A., Eisenstein, E. and Herzberg, O. (2003). Structure of the yibk methyltransferase from haemophilus influenzae (hi0766): a cofactor bound at a site formed by a knot. *Proteins: Structure, Function, and Bioinformatics*, **51** (1), 56–67.

Lindberg, M. O. e. a. (2006). Identification of the minimal protein-folding nucleus through loop-entropy perturbations. *Proceedings of the National Academy of Sciences*, **103** (11), 4083–4088.

Lua, R. C. (2012). Pyknot: a pymol tool for the discovery and analysis of knots in proteins. *Bioinformatics*, **28** (15), 2069–2071.

Macgregor, H. and Vlad, M. (1972). Interlocking and knotting of ring nucleoli in amphibian oocytes. *Chromosoma*, **39** (2), 205–214.

Madera, M. and Gough, J. (2002). A comparison of profile hidden markov model procedures for remote homology detection. *Nucleic Acids Research*, **30** (19), 4321–4328.

Mallam, A. L., Morris, E. R. and Jackson, S. E. (2008). Exploring knotting mechanisms in protein folding. *Proceedings of the National Academy of Sciences*, **105** (48), 18740–18745.

—, Rogers, J. M. and Jackson, S. E. (2010). Experimental detection of knotted conformations in denatured proteins. *Proceedings of the National Academy of Sciences*, **107** (18), 8189–8194.

Mansfield, M. L. (1994). Are there knots in proteins? *Nature Structural Biology*, **1** (4), 213.

Masek, W. J. and Paterson, M. S. (1980). A faster algorithm computing string edit distances. *Journal of Computer and System Sciences*, **20** (1), 18–31.

Michel, M., Skwark, M. J., Menéndez Hurtado, D., Ekeberg, M. and Elofsson, A. (2017). Predicting accurate contacts in thousands of pfam domain families using pconsc3. *Bioinformatics*, **33** (18), 2859–2866.

Micheletti, C., Di Stefano, M. and Orland, H. (2015). Absence of knots in known rna structures. *Proceedings of the National Academy of Sciences*, **112** (7), 2052–2057.

Miller, S., Janin, J., Lesk, A. M. and Chothia, C. (1987). Interior and surface of monomeric proteins. *Journal of Molecular Biology*, **196** (3), 641–656.

Mirny, L. A. (2011). The fractal globule as a model of chromatin architecture in the cell. *Chromosome Research*, **19** (1), 37–51.

Miyazawa, S. and Jernigan, R. L. (1996). Residue–residue potentials with a favorable contact pair term and an unfavorable high packing density term, for simulation and threading. *Journal of Molecular Biology*, **256** (3), 623–644.

Morcos, F., Jana, B., Hwa, T. and Onuchic, J. N. (2013). Coevolutionary signals across protein lineages help capture multiple protein conformations. *Proceedings of the National Academy of Sciences*, **110** (51), 20533–20538.

—, Pagnani, A., Lunt, B., Bertolino, A., Marks, D. S., Sander, C., Zecchina, R., Onuchic, J. N., Hwa, T. and Weigt, M. (2011). Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, **108** (49), E1293–E1301.

Moult, J., Pedersen, J. T., Judson, R. and Fidelis, K. (1995). A large-scale experiment to assess protein structure prediction methods. *Proteins: Structure, Function, and Bioinformatics*, **23** (3), ii–iv.

Najafi, S. and Potestio, R. (2015). Folding of small knotted proteins: Insights from a mean field coarse-grained model. *The Journal of Chemical Physics*, **143** (24), 12B606_1.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, **48** (3), 443–453.

Némethy, G. and Scheraga, H. A. (1977). Protein folding. *Quarterly Reviews of Biophysics*, **10** (3), 239–352.

Niemyska, W., Dabrowski-Tumanski, P., Kadlof, M., Haglund, E., Sułkowski, P. and Sulkowska, J. I. (2016). Complex lasso: new entangled motifs in proteins. *Scientific Reports*, **6**, 36895.

Noel, J. K. and Onuchic, J. N. (2012). *The Many Faces of Structure-Based Potentials: From Protein Folding Landscapes to Structural Characterization of Complex Biomolecules*, Boston, MA: Springer US, pp. 31–54.

Rawlings, C., Taylor, W., Nyakairu, J., Fox, J. and Sternberg, M. J. (1985). Reasoning about protein topology using the logic programming language prolog. *Journal of Molecular Graphics*, **3** (4), 151–157.

Reidemeister, K. (1927). Elementare begründung der knotentheorie. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, Springer, vol. 5, pp. 24–32.

Rolfsen, D. (1976). Knots and links. In *Mathematical Lecture Series 7*, Berkeley, Ca.: Publish or Perish.

Salisbury, F. B. (1969). Natural selection and the complexity of the gene. *Nature*, **224** (5217), 342.

Scalley-Kim, M., Minard, P. and Baker, D. (2003). Low free energy cost of very long loop insertions in proteins. *Protein Science*, **12** (2), 197–206.

Schneider, T. D. and Stephens, R. M. (1990). Sequence logos: a new way to display consensus sequences. *Nucleic Acids Research*, **18** (20), 6097–6100.

Schug, A., Weigt, M., Onuchic, J. N., Hwa, T. and Szurmant, H. (2009). High-resolution protein complexes from integrating genomic information with molecular simulation. *Proceedings of the National Academy of Sciences*, **106** (52), 22124–22129.

Sieradzan, A. K., Krupa, P., Scheraga, H. A., Liwo, A. and Czaplewski, C. (2015). Physics-based potentials for the coupling between backbone-and side-chain-local conformational states in the united residue (unres) force field for protein simulations. *Journal of Chemical Theory and Computation*, **11** (2), 817–831.

Sievers, F. and Higgins, D. G. (2014). Clustal omega, accurate alignment of very large numbers of sequences. In *Multiple sequence alignment methods*, Springer, pp. 105–116.

Sippl, M. J. (1990). Calculation of conformational ensembles from potentials of mena force: an approach to the knowledge-based prediction of local structures in globular proteins. *Journal of Molecular Biology*, **213** (4), 859–883.

Skolnick, J. and Kolinski, A. (1991). Dynamic monte carlo simulations of a new lattice model of globular protein folding, structure and dynamics. *Journal of Molecular Biology*, **221** (2), 499–531.

Skwark, M. J., Croucher, N. J., Puranen, S., Chewapreecha, C., Pesonen, M., Xu, Y. Y., Turner, P., Harris, S. R., Beres, S. B., Musser, J. M., Parkhill, J., Bentley, S. D., Aurell, E. and Corander, J. (2017). Interacting networks of resistance, virulence and core machinery genes identified by genome-wide epistasis analysis. *PLoS Genetics*, **13** (2), e1006508.

Smith, T. F., Waterman, M. S. *et al.* (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, **147** (1), 195–197.

Sriramoju, M. K., Chen, Y., Lee, Y.-T. C. and Hsu, S.-T. D. (2018). Topologically knotted deubiquitinases exhibit unprecedented mechanostability to withstand the proteolysis by an aaa+ protease. *Scientific reports*, **8** (1), 7076.

Steinegger, M., Meier, M., Mirdita, M., Voehringer, H., Haunsberger, S. J. and Soeding, J. (2019). Hh-suite3 for fast remote homology detection and deep protein annotation. *bioRxiv*, p. 560029.

Stevens, T. J., Lando, D., Basu, S., Atkinson, L. P., Cao, Y., Lee, S. F., Leeb, M., Wohlfahrt, K. J., Boucher, W., O'Shaughnessy-Kirwan, A. *et al.* (2017). 3d structures of individual mammalian genomes studied by single-cell hi-c. *Nature*, **544** (7648), 59.

Sułkowska, J. I., Morcos, F., Weigt, M., Hwa, T. and Onuchic, J. N. (2012a). Genomics-aided structure prediction. *Proceedings of the National Academy of Sciences*, **109** (26), 10340–10345.

Sulkowska, J. I., Niewieczerzal, S., Jarmolinska, A. I., Siebert, J. T., Virnau, P. and Niemyska, W. (2018). Knotgenome: a server to analyze entanglements of chromosomes. *Nucleic Acids Research*, **46** (W1), W17–W24.

Sułkowska, J. I., Rawdon, E. J., Millett, K. C., Onuchic, J. N. and Stasiak, A. (2012b). Conservation of complex knotting and slipknotting patterns in proteins. *Proceedings of the National Academy of Sciences*, **109** (26), E1715–E1723.

—, Sułkowski, P., Szymczak, P. and Cieplak, M. (2008). Stabilizing effect of knots on proteins. *Proceedings of the National Academy of Sciences*, **105** (50), 19714–19719.

Sumners, D. W. (1995). Lifting the curtain: using topology to probe the hidden action of enzymes. *Notices of the American Mathematical Society*, **42** (5), 528–537.

Taketomi, H., Ueda, Y. and Gō, N. (1975). Studies on protein folding, unfolding and fluctuations by computer simulation: I. the effect of specific amino acid sequence represented by specific inter-unit interactions. *International Journal of Peptide and Protein Research*, **7** (6), 445–459.

Taylor, W. R. (2000). A deeply knotted protein structure and how it might fold. *Nature*, **406** (6798), 916.

— (2007). Protein knots and fold complexity: some new twists. *Computational Biology and Chemistry*, **31** (3), 151–162.

— and Hatrick, K. (1994). Compensating changes in protein multiple sequence alignments. *Protein Engineering, Design and Selection*, **7** (3), 341–348.

— and Sadowski, M. I. (2011). Structural constraints on the covariance matrix derived from multiple aligned protein sequences. *PLoS One*, **6** (12), e28265.

Tkaczuk, K. L., Dunin-Horkawicz, S., Purta, E. and Bujnicki, J. M. (2007). Structural and evolutionary bioinformatics of the spout superfamily of methyltransferases. *BMC Bioinformatics*, **8** (1), 73.

Tubiana, L., Polles, G., Orlandini, E. and Micheletti, C. (2018). Kymoknot: A web server and software package to identify and locate knots in trajectories of linear or circular polymers. *The European Physical Journal E*, **41** (6), 72.

Wallin, S., Zeldovich, K. B. and Shakhnovich, E. I. (2007). The folding mechanics of a knotted protein. *Journal of Molecular Biology*, **368** (3), 884–893.

Webb, B. and Sali, A. (2014). Protein structure modeling with modeller. In *Protein Structure Prediction*, Springer, pp. 1–15.

Weigt, M., White, R. A., Szurmant, H., Hoch, J. A. and Hwa, T. (2009). Identification of direct residue contacts in protein–protein interaction by message passing. *Proceedings of the National Academy of Sciences*, **106** (1), 67–72.

Yeates, T. O., Norcross, T. S. and King, N. P. (2007). Knotted and topologically complex proteins as models for studying folding and stability. *Current Opinion in Chemical Biology*, **11** (6), 595–603.

Zschiedrich, C. P., Keidel, V. and Szurmant, H. (2016). Molecular mechanisms of two-component signal transduction. *Journal of Molecular Biology*, **428** (19), 3752–3775.